(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 May 2001 (17.05.2001)

PCT

(10) International Publication Number
WO 01/35617 A2

(51) International Patent Classification[7]: H04M 3/00

(21) International Application Number: PCT/US00/41354

(22) International Filing Date: 20 October 2000 (20.10.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/430,378 29 October 1999 (29.10.1999) US

(71) Applicant *(for all designated States except US)*: **TELERA** [US/US]; 910 East Hamilton Avenue, Suite 200, Campbell, CA 95008 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **UPPALURU, Prem** [—/US]; 11853 Shasta Spring Court, Cupertino, CA 95014 (US). **SUNDARAM, Mukesh** [—/US]; 5364 Gower Drive, San Jose, CA 95118 (US).

(74) Agents: **MALLIE, Michael, J.** et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th Floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 01/35617 A2

(54) Title: VIRTUAL INTELLIGENT NETWORK FOR USER INTERACTION SERVICES

(57) Abstract: A Virtual Intelligent Network (VIN) featuring a distributed call center system capable of answering, servicing, queuing and routing of calls at local point of presence to reduce communications costs and enhance operational efficiency for inbound and outbound call centers. The VIN includes a set of point-of-presence call center gateways distributed at points of presence close to the point of call origination that are connected by a user controlled virtual private network to premises call center gateways at business locations where the call centers reside.

VIRTUAL INTELLIGENT NETWORK FOR USER INTERACTION SERVICES

RELATED APPLICATION

The present application is related to and is a continuation-in-part of co-pending Application No. 09/249,395, entitled "Point-of-Presence Call Center Management System", filed on February 12, 1999, by Prem Uppaluru and Mukesh Sundaram, and assigned to the Assignee of the present invention.

FIELD OF THE INVENTION

The present invention relates to the field of telecommunications, and more particularly to the management of telephone calls to and from call centers.

BACKGROUND

Figure 1 is a functional diagram of a business call center connecting an end user 116 to a business call center 108 via an originating Local Public Switched Telecommunications Network (PSTN) 106, a Long Distance Network 114 and a terminating Local PSTN 106. Business call centers are typically put together by integrating multiple system components into a complete business solution to answer, service, queue and route inbound customer calls within the call center. These system components can include a Private Branch Exchange (PBX) 102, an Automatic Call Distributor (ACD) 112 and an Interactive Voice Response (IVR) System 110 in addition to customer service or help desk applications for the call center agents 104. Many call centers also deploy a Computer Telephone Integration (CTI) server 118 for providing intelligent call routing.

Traditionally, multiple vendors supplied the system components and system integrators pulled the components together into a solution and programmed them to implement business-dependent logic in servicing a call. However, to service the calls this traditional solution requires bringing the calls to the call center over long distance networks 114, provided by such companies as American Telephone & Telegraph (AT&T), Microwave Communications Incorporated (MCI) WorldCom and Sprint. This method added to the cost of operation of the call center 108, because as soon as a call was brought to the call center, an accumulation of long distance charges started. Furthermore, businesses with multiple physical locations were required to maintain programmable intelligence and develop multiple applications to implement the logic of servicing calls at every physical location, because of the insufficient flexibility of the long distance network that connects the locations.

Figure 2 is a functional diagram of a network-based call center connecting an end user 116 to a business call center 108 via an originating Local PSTN 106, a Long Distance Network 114 and a terminating Local PSTN 106. Network call centers may include a Switch 122, an ACD 112 and an IVR 110 within the Long Distance Network 114 to provide call answering, servicing and queuing services. Network call centers usually integrate carrier-provided voice

telecommunications with premises-based switching, call distribution, interactive voice response and computer telephony technologies to perform such services. Traditionally, network-based call centers relied on vender-specific proprietary systems and software to develop and deploy call management applications to perform answering, servicing and queuing services. Recently, telecom carriers have started offering value-added call management applications as managed network services. However, such managed network services are usually proprietary to the providing carrier, often insufficient in their flexibility and typically incomplete in functionality.

SUMMARY OF THE INVENTION

In accordance with one embodiment of the present invention, a telephone call to a first call center is directed to a second call center on the command of user specified applications. Under the direction of the user specified applications the call is serviced at the second call center while the user specified applications direct the first call center to establish a connection therein. When the connection in the first call center is established, the user specified applications direct the second call center to transfer or bridge the call with a telephone connection in the first call center via a telecommunications network.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings in which like references indicate similar elements and in which:

Figure 1 is a schematic diagram illustrating a conventional call center configuration with PBX, ACD and IVR systems located at the call center;

Figure 2 is a schematic diagram illustrating a conventional network-based call center configuration with Switch, ACD and IVR systems located within a long distance network;

Figure 3 is a flow diagram illustrating one example of an operational method of a Virtual Intelligent Network in accordance with an embodiment of the present invention;

Figure 4 is a flow diagram illustrating one process for selection of a premises call center in accordance with an embodiment of the present invention;

Figure 5 is a call flow diagram illustrating various aspects of the use of XML commands within a virtual intelligent network (VIN) in accordance with a n embodiment of the present invention;

Figure 6 illustrates examples of XML tags that may be included in an XML Page for transmission within a VIN in accordance with an embodiment of the present invention;

Figure 7 is a call diagram illustrating various aspects of the case of XML commands within a VIN for making outbound calls from a call center in accordance with an embodiment of the present invention;

Figure 8 is a schematic diagram illustrating components of a Virtual Intelligent Network according to an embodiment of the present invention that includes a point-of-presence (POP) Call Center Gateway, a Premises Call Center Gateway, a Call Center Network, a Network Operations Center and a Web Application Server;

Figure 9 is a schematic diagram illustrating a Virtual Intelligent Network configuration according to an embodiment of the present invention that includes a POP Call Center Gateway connected to a Premises Call Center Gateway over a Call Center Network controlled by user specified applications deployed on a Web Applications Server;

Figure 10 is a schematic diagram illustrating components of a Virtual Intelligent Network according to an embodiment of the present invention that supports a single business with multiple call center sites connected with multiple POP Call Center Gateways;

Figure 11 is a schematic diagram illustrating components of a Virtual Intelligent Network according to an embodiment of the present invention that supports multiple business call centers connected to multiple POP Call Center Gateways.

Figure 12 illustrates an exemplary network topology according to an embodiment of the present invention.

DETAILED DESCRIPTION

Although the present invention is described below by way of various embodiments that include specific structures and methods, embodiments that include alternative structures and methods may be employed without departing from the principles of the invention described herein. In general, the embodiments described below feature a Virtual Intelligent Network (VIN) that provides a distributed call center system controlled by a call center service provider, henceforth a user, and is capable of answering, servicing, queuing and/or routing calls at local points of presence to reduce communications costs and enhance operational efficiency for call centers. A call center is a single point of contact where telephone calls, may it be enquiries, order-placing, after-sales support or other call center activities, are handled and processed by a team of call center agents. These call center agents or operators can be based at a single site or at multiple sites, depending on the configuration of the call center and the organization that it belongs to. The industries and organizations where the use of call centers have become most widespread are those where there is a focus on customer service and a high volume of dispersed customer contact. These include banking and finance institutions, insurance companies, airline, hotels, and car rental organizations, and travel services.

Before describing embodiments of the present invention in detail, it is helpful to discuss some of the components of the VIN and other general concepts on which the present invention is based. The present invention utilizes point-of-presence (POP) call center gateways. POP call center gateways were described in detail in the above-cited related application (which is hereby incorporated by reference) and their applicability in efforts to minimize telephone charges for call center service providers was highlighted therein. A POP call center gateway is configured to answer, service, queue and/or route calls at a local point-of-presence, close to the point of call origination. POP call center gateways may thus provide initial processing of incoming calls; holding and queuing the calls until live operators at the called business call center are available. Once live operators become available, the POP call center gateways route the locally queued calls to the operators at the business call center. Thus, one benefit of using such POP call center

gateways is that the business call center service provider will only incur long distance charges for the time that a caller is actually connected to a business call center operator. Time spent on hold, in a queue or answering automated menu questions, etc., will not add to the long distance charges, because that portion of the call is treated as a local call, terminating at the POP call center gateway.

Another integral component of the present invention is a computer server. Servers are computer programs, which may be configured to execute on any one or more of a variety of computer hardware platforms to provide some service to other programs, called clients (which may also operate on a variety of computer platforms). A client and server communicate by means of message passing often over a network, and use some protocol, a set of formal rules describing how to communicate (i.e., transmit and receive) data, to encode the client's requests and/or responses and the server's responses and/or requests. A server may run continually, waiting for a client's requests and/or responses to arrive, or it may be invoked by some higher level software application, such as a continually running server, which controls a number of specific servers. Client-server communication is analogous to a customer (client) sending an order (request) on an order form to a supplier (server), who in turn dispatches the requested goods and an invoice (response). The order form and invoice (e.g., the terms and conditions specified therein) are part of the protocol used to communicate the request and response.

As further explained below, the present VIN employs a so-called eXtensible Markup Language (XML), a computer-based language designed to enable the use of the Standard Generalized Markup Language (SGML) on the World Wide Web (sometimes know as the "Web"). SGML is the international standard for defining descriptions of the structure and content of different types of electronic documents. Furthermore, the VIN utilizes the well-known Hypertext Transfer Protocol (HTTP), designed for distribution of hypertext documents over the Internet, to facilitate communication between network components (e.g., clients and servers) via XML messages over virtual private networks (VPN), which provide a tunnel within an otherwise public network for secure and private data communications.

In at least one embodiment of the present invention, the VIN includes a set of POP call center gateways, distributed at points-of-presence close to the point of call origination, interconnected by a user-controlled virtual private network to premises call center gateways, distributed at locations where the call centers exist. The POP and premises gateways, according to embodiments of the invention, are programmable telephony applications gateways that are dynamically controlled by call flow management (CFM) and interactive voice response (IVR) web applications. A CFM application uses information such as time of day, day of month, holidays, overflow, percentage allocation across multiple sites and agent group sets to select an answering resource to which to route a call. An IVR application may be regarded as a software application that uses a prerecorded database of voice (or synthesized voice) messages to present and/or collect user entered digits, interact with office databases and provide results the user. An IVR application may also optionally use speech recognition to secure input options from a user. These applications, CFM and IVR, can reside on web application servers, such as Netscape Application Server 2.1 and NetDymanics 4.0, potentially hosted at external Internet data centers and/or the call center premises and

connected to the same VPN as the POP and premises gateways. The CFM and IVR applications control the call flow and voice response behavior of these gateways through XML messages using HTTP and virtual private networks for distributed call resources (hardware and/or software) management.

The process of operating the VIN may be generally described with reference to the simplified flow diagram shown in **Figure 3.** In response to an inbound call being placed, at step 200 the user specified CFM and IVR applications direct a POP gateway, at or near the point of call origination, to intercept the inbound call. Thus, the POP gateway answers and terminates the inbound call locally, taking advantage of the economics of local interconnection, which generally provide for minimal access charges. In addition, the CFM, at step 204, issues a command (e.g., using an XML Page as described further below) to the POP call center gateway to provide a service, such as an interactive voice response-based automated service to hold and queue the call until operators are available to service the call, and/or to play music or customized announcements to the caller while the call is being held.

In parallel, the CFM, at step 208, may issue a command (again via an XML Page) to a premises call center gateway to originate a proxy call to a call center operator in its automatic call distributor (ACD). In response, the premises call center gateway generates and presents the proxy call to the ACD. The CFM command may also direct the premises call center gateway to monitor the progress of the proxy call within the ACD for operator availability and communicate the same to the CFM, at step 210. When the proxy call reaches the head of the premises call center queue and is about to be answered by a live call center agent, the premises call center gateway alerts the CFM. Based on this real–time information regarding the availability of the selected operator, the CFM application, at step 212, issues a command to the POP call center gateway to transfer the on-hold call to the premises call center gateway and also instructs the premises call center gateway to bridge the incoming call from the POP gateway to the proxy call (now about to be answered by the operator) in the premises ACD. Such local queuing and just-in-time call delivery saves on long distance communications charges that would otherwise be incurred if the call were to be earlier connected to the premises call center for queuing.

The process of selecting of a premises call center is generally described with reference to the flow diagram shown in **Figure 4.** At step 216, an incoming call is directed to an IVR application that is preferably developed as a web application and deployed on a web server. Such an application can be used to gather caller specific information (e.g., an account number, which the CFM application can use to intelligently route the call to the best matching answering resource. At step 218, the CFM directs a POP call center local to the incoming call to intercept the call and to provide interactive voice response-based automated services as described above. The CFM, at step 220, instructs the premises call center gateway selected as the best answering resource to insert a proxy call in its ACD. If the best answering resource is not available to service the call and the call needs to be transferred to a second best location, the CFM, at step 224, issues a command to the best answering resource to drop the proxy call from its ACD and directs, at step 228, the call center at the second best answering location to place a proxy call in its ACD. (This

process can be repeated as necessary (and as resources permit) until an answering resource capable of handling the call is located.) The second location, in response to the CFM command, monitors the progress of the proxy call, at step 230, and communicates operator availability to the CFM. Upon receiving a message from the second location that a live operator is available to service the call, the CFM, at step 232, instructs the POP call center gateway to transfer the call to the second location. Such local queuing at the POP call center gateway saves on long distance communications charges that would otherwise be incurred if the call were to be earlier connected to the best answering resource and then transferred to the second location.

When the call is ultimately connected to an agent at a premises call center, the VIN may take advantages of conventional Internet technologies such as IP telephony, Media Gateway Control Protocol (MGCP), Gatekeeper protocols, and virtual private networks with guaranteed quality of service for long distance voice communications to ensure a quality user experience is provided. A selected call center agent at the premises call center then answers the transferred call and provides customer service to/for the caller. Finally, when the caller or the call center agent hangs up the call, the CFM instructs the POP and Premises gateways to terminate the call.

Preferably, the POP and Premises gateways as well as the controlling CFM and IVR application servers, according to embodiments of the invention, are managed from a central Network Operations Center (NOC) 156 (see **Figure 9**) connected to the same VPN as the gateways and servers. Using the Simple Network Management Protocol (SNMP) and network management web tools, the NOC is capable of discovering, configuring, monitoring and managing all the components in the VIN. In addition, the NOC hosts other operations support systems including service provisioning, user billing, user support and other service management systems enabling the VIN to offer a suite of end-to-end managed user interaction services.

The present VIN not only gives a user (i.e., a call center provider) control over the inbound operations of the call center, but also over outbound call center operations. Under the guidance of the CFM application, the VIN can be used to dial specified long distance numbers as local numbers, filter out unanswered or machine-answered calls, play specified announcements or messages when calls are answered, and allow the called party to conduct interactive transactions with live operators and/or sotred interactive software applications. The CFM web application can specify a set of alternative long distance numbers and instruct elements of the VIN, via XML messages, to contact a called party and deliver prescribed voice messages. The VIN dispatches the request to attempt such contact to the appropriate POP gateway where the dialed number would be a local call. If the call is unanswered or answered by a machine, other specified numbers may be tried in a sequence until a call can be completed to a live person. When a call is answered, an IVR application can authenticate the called party and deliver the CFM-specified voice message or announcement and receive an acknowledgement from the called party. The IVR can also provide the called party with an option to be connected to a servicing agent. Upon the called party's selection of this option, the CFM directs the remote call center to establish a connection with the servicing agent, by requesting a proxy call in the remote call center's ACD

and monitoring the progress of the proxy call. When the connection with the servicing agent is established, the CFM instructs the remote call center to bridge the proxy call to the local call center, thus eliminating unnecessary long distance charges that would otherwise be accumulated transferring and queuing the call.

A more general view may thus regard the programmable VIN as a collection of intelligent, telephony ports, capable of originating or terminating calls, bridging call pairs and attaching callers to IVR applications. The programmability of the telephony ports is accomplished through the use of XML Pages, where each of the "tags" of the Pages instructs the telephony ports to perform designated call management functions. Such XML tags may be part of a general Voice Response Control Language (VCRL) for remotely controlling the behavior of POP call-center gateways. Thus, the set of point-of-presence call center gateways distributed at points of presence close to the point of call origination are interconnected by the programmable VIN to premises voice response servers at business locations where the call centers reside. The Call Flow Manager (CFM) orchestrates the creation and management of telephone calls in the VIN, under the control of the call center IVR applications.

As indicated above, the VRCL provides the syntax for the XML Page objects, which are generated at the CFM and used by a POP gateway to execute actions on behalf of the call center. The XML Page objects may be simple pages of ASCII text that are stored in a Web server's directory or generated by scripts at the server hosting the CFM. The XML Pages are transmitted by the CFM in response to HTTP requests made by a client application running on a POP gateway.

To more fully appreciate how the VRCL may be used by an IVR application running at a call center to control the actions of a POP gateway on behalf of the call center consider the following example with respect to the call flow diagram shown in **Figure 5**. When an inbound call arrives at a POP gateway, a data communication path is established between the POP gateway and the CFM. Soon after the call is accepted, the CFM transfers control to the IVR application for providing voice prompts and menu selection choices to the caller. The POP gateway fetches an initial XML Page from the IVR application, parses and executes the commands in the XML Page, and then requests the next XML Page from a location defined by a Universal Resource Locator (URL) specified in the first XML Page. This process then repeats, with the POP gateway continuing to get new sets of commands (represented by the tags in the XML Pages) from the IVR application. An important thing to note about VRCL is that unlike HTML a VRCL page represents a linear sequence of instructions. An HTML page is generally presented to a user all at once, but a VRCL page is audibly rendered and acted upon in a top to bottom sequence.

An IVR application may lead a POP gateway through a series of XML Pages depending upon the complexity of the application. For example, a series of inter-linked menus may need to be navigated by a caller before the caller's intentions and/or choices are fully defined. Ultimately, the IVR application may either terminate the call or transfer the call to a live operator. In either

case, the IVR application transfers control back to the CFM, which then directs the POP gateway to either setup a call over the PSTN or to terminate the call. The set of XML tags specified in the call flow shown in **Figure 5** may be understood with reference to the following discussion regarding examples of the type of tags which may be used in the VIN.

The VRCL further provides the ability to queue calls before being transferred to a live operator, and to integrate other Web-enhanced telephony applications, such as Click-to-Talk application. This is made possible by the use of dedicated XML tags, which can be interpreted by POP gateway to implement the desired commands specified therein. Thus, the VRCL defines a number of tags or elements (analogous to commands) to control the behavior of the POP servers. Figure 6 shows pictorially examples of the types of tags that can be included in an XML Page that is generated by an IVR application.

In the description below, the term "conversation controller" refers to the end point within the POP server that manages interaction with the IVR application or a CFM. As shown, the XML Page element is the root element for every XML Page used by the CFM and/or IVR for communication with the POP gateway. The Page element can have the following attributes (among others):

TYPE	An IVR application should use the value "IVR" while other applications may use different values (e.g. a Call Flow Manager may use the value "CC") to identify the source of the page.
CUSTID	A unique identifier for the call center.
PAGEID	(optional) A character string (can be numerals) to identify each page.
SESSIONID	An identifier for the particular session with the caller. This may be part of a query-string that is sent by a POP gateway in the HTTP request for each page.
HREF	The URL of the next XML Page to fetch if the end of the page is reached (a child element may transfer control to another URL from somewhere within the page).

Each XML Page represents a unit of work (i.e., a work order) to be performed by the POP gateway. The unit of work may consist of a single action or it may consist of a linear list of several actions. At the top-level, basic elements (e.g., PLAY, TONEMAP, VOICEMAP and EXCEPTIONMAP), complex elements (e.g., MENU, FORM), and/or anchor elements for

labeling (e.g., A) may be included in a Page. In addition a RECFILE element for handling voice recordings and/or a SET element for setting user-defined variables may be provided (not shown in the diagram). The sub-elements of an XML Page root are oriented towards call flow, interaction with a caller or call control.

Some possible children (sub-elements) of an XML Page root element that are oriented towards call flow and interaction management are: PLAY, TONEMAP, VOICEMAP, EXCEPTIONMAP, MENU, FORM, A, RECFILE, and SET. Sub-elements oriented towards call control are: CREATE_LEG_AND_DIAL, HANGUP_AND_DESTROY_LEG, QUEUE_CALL, QUEUE_DROP, BRIDGE_CALL, UNBRIDGE_CALL, LEG_WAIT, ALERT_LEG, and END_SESSION. These sub-elements can occur in any order and multiple times. The CFM is able to utilize a larger set of more primitive call control tags which cannot be issued by an IVR application. This is because the CFM is a VIN supervisory process.

The PLAY element allows an IVR application to direct a POP gateway server to play a voice file, pause for a specified amount of time, play out a number or set of digits, and so on. It can have the following attributes:

INTERRUPTIBLE	(optional) YES or NO. By default a message being played is interruptible by a tone input. However, it can be made non-interruptible by using a value of NO (except in a MENU - see below).

PLAY has several possible children elements, which can be used in any order, any number of times: VOICE, PAUSE, STRING, DATE, TIME, NUMBER, CURRENCY, ORDINAL, INDEXFILE, and DTMF.

The VOICE element specifies the voice file to be played. The voice file can be any computer-readable audio file, such as a .vox file or a .wav file. In one embodiment the VOICE element has two attributes, one of which (SRC) is required and the other of which (TEXT) is optional.

SRC	The fully specified URL of the file to be played.
TEXT	(optional) A textual rendering of what is in the SRC file.

VOICE has no children and is an empty element (i.e., the closing tag can be a simple"/>" at the end of the attributes).

The PAUSE element specifies a period of silence. It may, for example, follow a VOICE tag within a PLAY element. If its parent PLAY tag is not part of a MENU or a FIELD, the PAUSE merely introduces silence for the specified TIME. On the other hand, if its parent PLAY

tag is part of a MENU or FIELD, the PAUSE works as a timeout period, during which time the caller is expected to enter at least one digit (see also the description of MAX_IDD -- max inter-digit-delay -- in the INPUT tag description below).

PAUSE has two attributes: TIME and UNIT:

TIME	The value of the time-period for which silence is desired, and during which caller input is desired (if part of a MENU or FIELD).
UNIT	(optional) The units (SEC or TENTHS) in which the TIME is specified. TENTHS means unit = 0.1 SEC.

PAUSE has no children.

The STRING element, which has no children, is used to indicate that the IVR application wants the POP server to play out a string of letters, digits and/or special characters, one at a time. It has two attributes, both of which are required.

SRC	The URL of the index file (see below) to be used by the voice subsystem for speaking out the characters.
VALUE	The string of letters/digits/characters to be played out. The string can contain any combination of the following symbols: A,B,C,D,E,F,G,H,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y, Z,0,1,2,3,4,5,6,7,8,9, +,<,=,%,-,>,&,.,#,*,@, etc. (i.e., any ASCII character). Note, because of the special meaning that conventional XML parsing code attaches to "<", ">", and "&" symbols, these may need to be actually written as "<", ">" and "&" respectively.

As an example of how the STRING tag may be used, consider VALUE = "IBM". When played out, this will be rendered as "Aiyee Bee Emm". If VALUE = "3+4", the phrase played out will be "Three Plus Four", and so on. A special index file to be used when playing out STRING, DATE, TIME, NUMBER, CURRENCY, and ORDINAL data may store sounds for all the common prompts used in playing out these types of data. It may also include a header section in the beginning of the file that is an array of offsets pointing, in a well defined order, to the starting points of the prompts therein. The format of the index file is not critical to the present invention.

The DATE element is used to indicate that the IVR application wants the POP server to play out a date to the caller. It has three attributes, all of which are required.

SRC	The URL of the index file to be used by the voice subsystem for speaking out the date (see the above description of the STRING tag).
VALUE	The value of the date to be spoken out. The value may be specified as yyyymmdd (8 digits) ,mmdd (4 digits), w:yymmdd(10 characters), w:mmdd (6 characters), or another user-defined format. w refers to the day of the week with 0 corresponding to Sunday, 1 to Monday and so on. yy, mm and dd refer to the year, month and day of the month, respectively.
FORMAT	This indicates how the date should be played out. Possible values of this attribute are DDMM, MMDD, DDMMYYYY ,MMDDYYYY, W:DDMM, W:MMDD, W:DDMMYYYYY and W:MMDDYYYY. For example, DDMMYYYY means that the date should be spoken out as "day - month - year". W:MMDD means that the date should be spoken out as (say) day of the week - month - day of the month.

DATE has no children.

The TIME element may be used to indicate that the IVR application wants the POP server to play out a time to the caller. It has three attributes, all of which are required.

SRC	The URL of the index file to be used by the voice subsystem for playing out the time (see the above description of the STRING tag).
VALUE	The value of the time to be played out. It may be

	specified either as hhmm (4 digits) or as hhmmss (6 digits), or another user-specified format (e.g., in 24 hr time fashion).
FORMAT	Describes how the time should be played out. Possible values are 12_HOUR or 24_HOUR.

TIME has no children

The NUMBER element may indicate that the IVR application wants the POP server to play out a number to the caller. It has two attributes, both of which are required.

SRC	The URL of the index file to be used by the voice subsystem for speaking out the number (see the above description of the STRING tag).
VALUE	The value of the number to be played out.

NUMBER has no children

The CURRENCY element may be used to indicate that the IVR application wants the POP server to play out a money value to the caller. It has three attributes, all of which are required.

SRC	The URL of the index file to be used by the voice subsystem for speaking out the currency (see the above description of the STRING tag).
VALUE	The value of the currency to be spoken out. It may or may not have a decimal point.
COUNTRY	3 character (e.g., ISO-4217 compliant) currency code.

CURRENCY has no children

The ORDINAL element may be used to indicate that the IVR application wants the POP server to play out an ordinal (e.g. "First", "Third" etc) to the caller. It has two attributes:

SRC	The URL of the index file to be used by the voice

subsystem for speaking out the ordinal.

VALUE	The value of the ordinal to be spoken out. The allowed values may be: 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19 ,20,21,22,23,24,25,26,27,28,29,30,31,40,50,60,7 0,80,90,100,1000,1000000,1000000000, etc.

ORDINAL has no children

The INDEXFILE element may be used by an IVR application for playing date, currency, digits etc., if the above-mentioned index file format is unsuitable for playing the desired prompts (e.g., the call center may use some additional prompts or the call center's index file may not conform to the above-described file type). In such a case the IVR application should not use the DATE, CURRENCY, etc., tags described above, but rather use the INDEXFILE tag and specify the offset and length within the file to ensure that the desired prompts are played out. INDEXFILE may have up to four attributes:

SRC	The URL of the call center's index file to be used by the voice subsystem for playing out the prompts.
OFFSETS	A list of comma-separated offsets, indicating the byte offsets for the starting points of the prompts to be played in the file. For example OFFSETS may have a value of: "0031545,0038040,0022060,0041122" for the phrase "MSFT, assuming that the offsets for the beginning of M, S, F and T prompts are 0031545, 0038040, 0022060 and 0041122, respectively.
LENGTHS	A list of comma-separated lengths, indicating the lengths of the prompts to be played in the file. There should be one-to-one correspondence between the OFFSETS and the LENGTHS.
TEXT	An optional textual rendering of what is in the prompts.

INDEXFILE has no children.

The DTMF tag should only be used as a child tag under PLAY. The purpose of this tag is to allow the IVR application to send an in-band DTMF tone on an already established call to an ACD/PBX. This allows the IVR application to send account numbers, etc., and take the ACD/CTI application to a stage where the caller will not have to re-enter/repeat information before talking to a live operator. The DTMF tag has only one attribute:

VALUE	The tones that are to be sent (e.g. "2690,,,*#,3396"). A comma in the value may be used to indicate a pause

A PLAY DTMF combination should not be used within a MENU so that no interference between tones within a MENU will result. DTMF has no children.

The TONEMAP element allows an IVR application to specify a mapping between tone keys entered by the caller and URLs of XML Pages to go to in response to the user input. A TONEMAP can exist either at the top level within a XML Page or it can be used as a sub-element of a MENU. In the first case (when specified outside a MENU), the TONEMAP has global effect in the sense that it is remembered across pages. In the second case (when specified as part of a MENU) its scope is local. Within a MENU the local TONEMAP takes precedence over the global TONEMAP for TONEs (see below) which are in conflict between the two maps. For TONEs in the global TONEMAP that do not have any conflict with a local TONEMAP, such TONEs will have the same effect as for the global case, even within the MENU. After a MENU is exited, the applicable TONEMAP reverts to the applicable global TONEMAP.

TONEMAP may have a TONE child element, which may be used to specify the mapping between one tone key and the URL from which to fetch the next page. TONE has the following attributes. The first (TONEID) is required. One out of the second or third attributes must also be entered.

TONEID	One of 0,1,2,3,4,5,6,7,8,9,*, #,OTHER. OTHER includes all the remaining keys left unspecified in the TONEMAP. (required)
HREF	The URL from which to fetch the next page. This can also point to a label within the current Page (see description of the A element, below). (optional in place of REPEAT)
REPEAT	Value must be MENU. This allows control to

	go back to the PLAY at the beginning of the MENU in case the associated tone key is entered. (optional in place of HREF)

TONE has no children

The VOICEMAP element allows an IVR application to specify a mapping between certain voice inputs received from a caller and URLs of XML Pages to go to in response thereto. A VOICEMAP can exist either at the top level within an XML Page or it can occur as a sub-element of a MENU. In the first case (when specified outside a MENU), the VOICEMAP has global effect in the sense that it is remembered across pages (e.g., any time a caller says "operator", a certain URL's XML Page may be accessed to transfer control to an operator). In the second case (when specified as part of a MENU) its scope is local. Within a MENU the local VOICEMAP takes precedence over the global VOICEMAP for PHRASEs (see below) which are in conflict between the two maps. For PHRASEs in the global VOICEMAP that do not have any conflict with the local VOICEMAP, they have the same effect as for the global casse, even within the MENU. After the MENU is exited, the applicable VOICEMAP reverts to the global VOICEMAP.

VOICEMAP may have a PHRASE child element, which specifies the mapping between one voice command (phrase) and the URL from which to fetch the next XML Page in response thereto. It has two attributes, both of which are required.

SRC	A voice recognition file or a voice grammar file.
HREF	The URL from which to fetch the next page. This can also point to a label within a page (see description of the A element, below).

PHRASE has no children

The EXCEPTIONMAP element may be used to specify a mapping between certain exceptions (for example a timeout) and the URLs of the XML Pages to fetch in the case of such an event. Like a TONEMAP, an EXCEPTIONMAP can be either global (when it occurs at the top level as a sub-element of an XML Page) or local (as a sub-element of FIELD or MENU) in scope. Within a FIELD or MENU the local EXCEPTIONMAP takes precedence over the global EXCEPTIONMAP for EXCEPTIONs (see below) which are in conflict between the two maps. For EXCEPTIONs in the global EXCEPTIONMAP that do not have any conflict with the local EXCEPTIONMAP, they still have the same effect as for the global case, even within the FIELD or MENU. After the FIELD or MENU is exited, the applicable EXCEPTIONMAP reverts to the global EXCEPTIONMAP.

The EXCEPTIONMAP may have EXCEPTION children elements, which specify the mappings between exceptions and the URL from which to fetch the corresponding pages, if such exceptions are encountered. EXCEPTION has three possible attributes: EVENT, which is required; HREF, which is required for a global EXCEPTIONMAP; and REPEAT, which may be used in place of HREF for an EXCEPTIONMAP within a MENU or FIELD.

EVENT	Possible values: TOOFEWDIGITS – applicable in FIELDs. The number of digits entered by the caller is less than expected. TIMEOUT – caller entered no input within a period specified by previous PAUSE tag CALLERHUP – caller unexpectedly hangs up OTHER – catch-all for unanticipated exceptions
HREF	The URL from which to fetch the next page in case of the exception specified by EVENT being encountered. This can also point to a label within the current page (see description of the A element below)
REPEAT	Value can be "MENU" or "FIELD" for an EXCEPTIONMAP specified within a MENU or FIELD. This allows control to go back to the PLAY at the beginning of the MENU/FIELD in case the EXCEPTION occurred.

An EXCEPTION can be raised at any time and it may be a completely asynchronous event (e.g., where the caller hangs up suddenly). Therefore it is useful to specify a default global

EXCEPTIONMAP in one of the very first XML Pages sent by the IVR application. EXCEPTION has no children.

The MENU element offers an IVR application the ability to have a POP server play a message and then collect a caller's selection (from a menu of choices) and fetch a new XML Page based on such selection. The caller can make his/her selection by either entering a tone key or speaking a word or phrase (if voice recognition is supported). MENU has one attribute:

MAXREPEATS	(optional) The maximum number of times control can return to the beginning of the MENU. To prevent infinite loops, control goes to the HREF specified in the XML Page tag specified at the top of the page if a REPEAT exception is encountered more than MAXREPEATS times in succession. Note, MAXREPEATS also applies to the case where control goes back to a FIELD tag by virtue of using an A (anchor) label.

MENU can have the following children: PLAY, TONEMAP, VOICEMAP, and EXCEPTIONMAP. One or both of TONEMAP and VOICEMAP should be present. When used in a MENU, PLAY should include a PAUSE sub-element to specify a timeout within which it expects the caller to enter a tone or provide a voice input. If the caller enters no tone/voice input within the timeout period, control may go to the HREF specified in the TIMEOUT EXCEPTION. If no TIMEOUT EXCEPTION is specified in the local EXCEPTIONMAP, the global EXCEPTIONMAP is consulted. If no TIMEOUT EXCEPTION exists even at the global level, then the next page as specified in the HREF attribute of the current XML Page is fetched. This same hierarchy of exception handling may be used for all exceptions. It is also possible to transfer control to an anchor/label on the current XML Page by using the A sub-element (see below) and specifying #label in the XML Page's HREF attribute.

A FORM element may be used to instruct the POP server to collect information (e.g., multi-key inputs or voice recordings) about a number of FIELDS and then send the information back as NAME, VALUE pairs to the HREF specified in the FORM.

A FORM can have the following attributes (only HREF is required):

HREF	The URL to send the information collected from the caller to. This URL will typically be a binary program or a perl, java or ASP script.

METHOD	(optional) GET or POST. Specifies whether the data is sent to the URL as a query-string or as standard input.
SRC	(optional) The URL of a file (e.g., a .vox file) to play while the data submitted in the FORM is being processed by the IVR application. The file will typically contain a message like "please wait while we process your input". If the SRC attribute is not included, the caller will hear silence while the POP server is waiting for the IVR application to send the next XML Page after processing the FORM data.
SUBMIT_PARTIAL	(optional) This specifies whether the FORM should be submitted to the IVR application if the caller hangs up after filling out some or all of the fields and does not remain on-line to interact with the IVR application. Possible values are: PARTIAL_OK: send values for whatever fields have been filled up COMPLETE_ONLY: send values only if all fields have been filled LAST_FIELD_HUP: send values if all the fields have been filled in and the caller is still interacting with IVR or the caller hangs up while entering data (or recording voice) for the last field
POSTHREF	(optional - for voice fields only). This is the URL of an application that will receive the voice recording files. The voice file that contains a message recorded by the caller is POSTed by the POP server at a later time, determined by a network manager. The manager (which may be a

	software application) ensures that not too much of the available network bandwidth is consumed in transporting the voice recording files. The POSTHREF URL should contain within the query string the following: ?SESSIONID=$sessionid$ & FIELDNAME=$fieldname$ & RESULT=$status$

A FORM may have one or more FIELD children, which allow the POP server to collect information (e.g., multi-key input or voice recordings) for a single named field according to certain rules specified in the INPUT sub-element. A FIELD has the following attributes:

HREF	(For VOICE only). The URL where the contents of the voice file are to be POSTED after the recording of the voice file is complete If no HREF is specified then the network manager will POST the files to the POSTHREF specified in the FORM.
MAXREPEATS	(optional) The maximum number of times control can go back to the beginning of the FIELD. To prevent infinite loops, control goes to the HREF specified in the XML Page tag specified at the top of the page, if a REPEAT exception is encountered more than MAXREPEATS times in succession. Note that MAXREPEATS also applies to the case where control goes back to the FIELD tag by virtue of using the A (anchor) label.

A FIELD should always be composed of a triplet consisting of the following sub-elements: PLAY, INPUT, and EXCEPTIONMAP

The INPUT element specifies the rules for the telephony interface at the POP server for collecting input for a FIELD. It can have the following attributes:

NAME	The name of the variable in which to collect the

	caller's input. For VOICE input, the value of NAME (in the NAME,VALUE pairs sent back by the FORM) is either the FILENAME returned in the RECFILE tag by the application, or is the URL of the recording file that has been stored locally on the POP server (see also HREF attribute description for FIELD tag). If POSTHREF is specified in the FORM the value of NAME (in the NAME,VALUE pairs sent back by the FORM) is the name of the recording file that has been stored locally on the POP server.
TYPE	(optional) TONES or VOICE or EITHER to indicate the type of expected input.
MINDIGITS	(Optional) Minimum number of digits required for the field (not valid for voice).
MAXDIGITS	(optional) Maximum number of digits that can be entered for this field (not valid for voice). Add 1 for the ACCEPT key if so specified.
ACCEPT	(optional) The tone key which signals the end of a user's input (e.g. "#").
MAX_IDD	(optional) The maximum interdigit delay in seconds, after which the end of a user's input is implied.
MAXSILENCE	(optional) During recording of voice input, the maximum silence (in seconds) before the end of a user's input is implied.
MAXRECLEN	(optional) During recording of voice input, the maximum length of the message in seconds, that can be recorded.
VOICEFILE_FORMAT	(optional) Desired format and sampling rate for

the recording file.

INPUT has no children.

If POSTHREF is specified in the FORM tag, then the IVR application will first receive an HTTP GET request with the NAME, VALUE pairs for the different fields as specified in the INPUT NAME attributes. The values will be the names of the recording file that has been stored locally on the POP server. If necessary, the IVR application may playback the contents of the recording file to the caller. Later, the IVR application will receive multiple HTTP POST requests with the binary data of the recorded files in the contents.

A RECFILE tag may be used by an IVR application in response to a POST request containing data for a VOICE field input. It has the following attribute:

FILENAME	Name of the file in which the POSTed voice data has been stored by the IVR application server.

RECFILE has no children.

The "A" (anchor) element may be provided as a labeling mechanism in an XML Page. A URL can specify a target within a current or different XML Page by using #name (as in HTML). This provides the ability to direct a POP server to start parsing and executing an XML Page not at the beginning of the page but at a labeled point within the page. It can also be used as a mechanism to jump to a certain point within a current XML Page. A has one required attribute and has no children.

NAME	The label name by which this anchor will be referred to by URLs.

The SET element allows an IVR application to set the value of a user-definable variable for later substitution by a POP server. It is a convenient mechanism to allow an IVR application to specify an association between a particular value and a particular variable name such that the POP server should substitute the variable name with the value when the IVR application uses the variable name later in the same or another XML Page. SET has two required attributes and no children.

VARNAME	The name of the variable.
VALUE	The value to assign to the variable.

SET thus offers a stateless IVR application a powerful mechanism to maintain state and use session variables for any Web server environment. The IVR application may define and set the value of a session variable (a variable that lasts during the life-time of a current session/call) by using the SET tag. The value of the session variable can then be later retrieved by the application by putting the variable in the query string of the application URL that needs to use the value of the variable.

Described below are the call control oriented tags that an IVR application may issue to the POP server. The CREATE_LEG_AND_DIAL element may be sent by an IVR application in order to make an outbound call. The attributes of CREATE_LEG_AND_DIAL are:

TELNUM	The telephone number that the telephony server needs to call.
IVRURL	URL from where the conversation controller of the new leg should fetch its first XML Page after the outbound call is made and the call is answered. This can be used by the IVR application to play an audio message or to carry out other VRCL commands. This attribute can also carry a value of LEG_WAIT, in which case the new conversation controller just goes into an interruptible wait state and the person answering the phone at the called number hears silence until an interrupt causes the conversation controller to bridge the call. The IVRURL is not used when the BRIDGE attribute is set to YES.
ANI	(optional) The ANI that the telephony manager should send in the outgoing call.
BRIDGE	(optional) YES or NO. If YES, the new call is bridged with a call on a current LEG as soon as the new call is dialed and answered.

The TELNUM attribute can contain commas in its value. A comma has two meanings inside TELNUM. The first comma (from the left) signifies that the digits to the left of the first comma will be used for dialing out and establishing the call, while the digits to the right of the first comma will be sent out as in-band DTMF tones on the established call. The comma(s) within

the digits to the right of the first comma signify that a pause should be inserted for each comma in the DTMF tone sequence. For example TELNUM="12159090,,,**,90061234#" instructs the telephony server to dial 12159090 for establishing the call, pause, send ** tones, pause and send 90061234# tones. If better control is needed in order to make sure that the call is answered before any tones are sent, PLAY DTMF tags should be used by sending the created leg to IVRURL and using these tags in the XML Page sent back by IVRURL. In that case, the BRIDGE=NO option should be used. Using commas in the TELNUM field provides a convenient mechanism for combining dialing out with fixed pauses and sending some in-band DTMF tones, but it does not provide as much control as the DTMF tag.

CREATE_LEG_AND_DIAL does not have any children. Normally a CREATE_LEG_AND_ DIAL tag will be followed by a LEG_WAIT tag (or a PLAY followed by a LEG_WAIT) in order to wait for the other leg to synchronize with this leg.

Upon receiving this tag, the telephony server sends a CREATE_LEG_AND_DIAL_REQ NextAction request to the CFM. The CFM reserves an outbound port on an appropriate telephony server (based on TELNUM), creates a new conversation controller on the telephony server and then sends a DIAL_CALL XML tag and (optionally) a BRIDGE_CALL tag to the new conversation controller to make an outbound call and bridge it to the current call. In the case of errors, the global EXCEPTIONMAP is consulted and the URL corresponding to CREATE_LEG_ERROR, DIAL_ERROR or BRIDGE_ERROR is contacted, depending upon the type of error.

The HANG_UP_AND_DESTROY_LEG element may be used by an IVR application to inform the POP server that the call on the designated leg should be terminated and the conversation controller (LEG) receiving this tag should be destroyed. The attributes of this element are:

REASON	(optional) NORMAL or ERROR or CALLERHUP.

This element does not have any sub-elements.

A QUEUE_CALL tag may transmitted by the IVR application to put a call on hold. The attributes of this tag are:

AGENTGRP	Identifies the agent group where call should be queued (e.g. SALES).
STATUS_INT_TIME	(optional) This is the interval at which the IVR application should be informed about the status of the queued call.

STATUS_INT_URL	(optional) This is the URL where the IVR should be informed about the status of the call on hold after receiving the CFC interrupt to give ICR/ACD status. (The QUEUE_ERROR exception in the global EXCEPTIONMAP will be consulted).
ACD_TELNUM	(optional) This is the outbound telephone number that needs to be dialed by the telephony server to queue the call with the ACD
ANI	(optional) This is the ANI that the IVR application wants the POP server to insert into the outbound call that will be placed to the ACD or ICR. This can be either a telephone number or the string ANI
USR_PARAMS	(optional) Information that the caller entered into the IVR system before requesting transfer to an operator. This field would contain sub-parameters and values separated by colons (":") with the different parameter-value pairs separated by commas e.g., PARAM1:25, PARAM2:busy, PARAM3:411. The names and values of the sub-parameters are translated by the ACD adapter for interpretation by the ACD.
AGENT_URL	(optional) This the URL from where the LEG of the telephony server with the outbound call to the ACD should get its XML Page after the AGENT is connected. This can be used for playing some caller-related information into the agent's telephone, before the call is bridged with

that of the caller.

QUEUE_CALL can optionally have an EXCEPTIONMAP as a child.

The QUEUE_DROP tag is sent by the IVR application when the caller and/or application decides that it is not worth continuing to hold the call and that the call should be dropped from the ACD queue. This tag has no attributes, but QUEUE_DROP can optionally have an EXCEPTIONMAP as a child.

The identification of the QUEUE to be dropped is implicit because there can be only one queue for a conversation. The telephony server sends a QUEUE_DROP_REQ to the CFM upon receiving this tag.

The BRIDGE_CALL tag is sent by the IVR application to bridge a current call (call on a current leg) with another call on the same POP server or a call on a different gateway on a different POP gateway. BRIDGE_CALL has one required attribute:

LEG_ID	The identification of the other leg to be bridged with the call on the current leg. A value of ALL will bridge the call on the current leg with all other calls associated with various legs of this session.

BRIDGE_CALL can optionally have an EXCEPTIONMAP as a child.

After the two calls are bridged, all bridged calls go into a LEG_WAIT state. If there is an error in bridging the calls, the BRIDGE_ERROR exception is raised and the next XML Page is fetched from the HREF specified in the exception.

An XML Page with the UNBRIDGE_CALL tag is sent by the IVR application in to break the bridge between various legs of a call. The attributes of UNBRIDGE_CALL are

LEG_ID	(required) Used to identify the other leg which is to be unbridged. A value of ALL will break the bridge between the current leg's call and all other calls bridged with the current call.
OTHER_LEG_URL	(optional) The URL from where the other leg(s) should fetch their next XML Page after being unbridged. If this attribute is omitted, the other leg(s) will continue in LEG_WAIT state and an ALERT_LEG tag will be needed to wake them up.

It is assumed that one of the legs is implicitly the one receiving this UNBRIDGE_CALL tag.

UNBRIDGE_CALL can optionally have an EXCEPTIONMAP as a child. After the calls are unbridged, the POP server executes the next tag following the UNBRIDGE_CALL tag or (if there is none) fetches the next XML Page from the HREF specified in the XML Page root element tag at the top of the page. If there is an error in unbridging the calls, the UNBRIDGE_ERROR exception is raised and the next XML page is fetched from the HREF specified in the exception.

An XMLPage with the LEG_WAIT tag is sent by the IVR application to put the conversation controller receiving this tag into a wait state. The conversation controller is then expected to do nothing except wait for an event from the telephony manager (e.g., caller hangup) or an interrupt from the CFM or an ALERT_LEG tag from the IVR application. LEG_WAIT has no attributes and no children.

The ALERT_LEG tag is used by the IVR to interrupt a conversation controller thread in a LEG_WAIT state and to instruct it to send an HTTP GET request to the URL specified as an attribute. The attributes of ALERT_LEG are:

LEG_ID	The identifier of the other leg(s) that is/are to be alerted. The legs to be alerted would normally be sitting in a LEG_WAIT state. If a value of ALL is used, all other legs of this session will be alerted.
IVRURL	URL from where to fetch the next XML Page for the other legs after they have been alerted.

ALERT_LEG has no children. This tag may be used, for example, after unbridging a call to alert the other leg and direct it to a specified URL.

An END_SESSION may be used by the IVR application to destroy all LEGS of a current session and hang up all the associated calls. This gets translated into a END_SESSION_REQ NextAction request which is sent by the telephony server to the CFM. Upon receiving such a request, the CFM then sends HANGUP_CALL and DESTROY_LEG tags to all the conversation controller threads associated with the session.

Some examples of the uses of the above tags may be helpful at this point. In the first example, Example 1, an XML Page for playing out a welcome message and setting a global TONEMAP and EXCEPTIONMAP is presented. First, the page type and customer identifiers are specified. Note that a call center server URL of "customer.com" is used as an address where the various pages are stored.

<!-- Example # 1 -->

<?xml version="1.0" ?>
<XMLPage TYPE= "IVR" CUSTID="customer" PAGEID="0001"
VERSION="1.7"
SESSIONID=" 3"
HREF="www.customer.com/tele/initial.asp?SESSIONID=3" >

Now, a global TONEMAP is defined. The TONEMAP allows a caller to press a 0 at any time and get through to an operator, or to press the * key and go straight to a login menu. Also, if the caller presses an invalid key, control transfers to another XML Page, located at the specified HREF, that handles such a case. This will apply to all subsequent pages unless over-ridden by a new TONEMAP.

<TONEMAP >
<TONE TONEID="0"
HREF="www.customer.com/tele/operator.asp?SESSIONID=3" />
<TONE TONEID="*"
HREF="www.customer.com/tele/login.asp?SESSIONID=3" />
<TONE TONEID="OTHER"
HREF="www.customer.com/tele/problem.asp?SESSIONID=3" />
</TONEMAP>

Now a global EXCEPTIONMAP can be defined. The EXCEPTION MAP causes an exception to be raised any time the caller fails to press a key within a specified timeout period when asked to enter a FIELD value or make a choice from a MENU. Control then transfers to the specified timeout exception handler (timeout.asp?SESSIONID=3). An exception is also raised if the caller hangs up suddenly. The HREF attribute in each EXCEPTION specifies which XML Page to fetch next in case such an exception happens. The global EXCEPTIONMAP applies to all subsequent pages unless over-ridden.

<EXCEPTIONMAP >
<EXCEPTION EVENT="TIMEOUT"
HREF="www.customer.com/tele/timeout.asp?SESSIONID=3" />
<EXCEPTION EVENT="CALLERHUP"
HREF="www.customer.com/tele/terminate.asp?SESSIONID=3" />
<EXCEPTION EVENT="OTHER"
HREF="www.customer.com/tele/error.asp?SESSIONID=3&ERROR=$last-error$
& DESCRIPTION=$last-error-
string$&URL=$error-url$ />
</EXCEPTIONMAP>

Now commands can be issued to play a greeting message. The message should not be interrupted by the caller and so there is an instruction to ignore any keys that he/she may press while the message is playing.

```
<PLAY        INTERRUPTIBLE="NO" >
<VOICE       SRC="www.customer.com/tele/greeting.vox"  TEXT="Welcome to
             Customer's Call Center" />
<PAUSE       TIME="1"  UNIT="SEC" />
</PLAY>
</XMLPage>
```

On reaching this end of the page, the next XML Page specified at the start of the initial Page will be fetched (initial.asp).

A second example, Example 2, provides an XML Page for announcing a set of choices and accepting a single tone input from a caller.

```
<!-- Example # 2 -->
<?xml        version="1.0" ?>
<XMLPage     TYPE= "IVR" CUSTID="customer"  PAGEID="v101"
VERSION="1.7"
             SESSIONID=" 3"
        HREF="www.customer.com/tele/custsvc.asp?SESSIONID=3" >
<A           NAME="label1" />
<MENU>
<PLAY>
<VOICE       SRC="www.customer.com/tele/initial.vox" TEXT="Press 1 if you have
             an account, Press 2 to open an account"  />
<PAUSE       TIME="5" />
</PLAY>
<TONEMAP >
<TONE        TONEID="1"
        HREF="www.customer.com/tele/login.asp?SESSIONID=3" />
<TONE        TONEID="2"
        HREF="www.customer.com/tele/newacc.asp?SESSIONID=3" />
<TONE        TONEID="OTHER"  HREF="#label1" />
</TONEMAP>
<EXCEPTIONMAP >
<EXCEPTION EVENT="TIMEOUT" HREF="#label1" />
</EXCEPTIONMAP>
</MENU>
```

</XMLPage>

With the global TONEMAP as specified on the previous Page (from Example 1), this Page's local TONEMAP will cause "0", "*", "1" and "2" to be all valid responses. After the MENU is exited, the TONEMAP specified within the MENU is no longer applicable and only "0" and "*" will be accepted as valid tones. Recall that any global map (EXCEPTIONMAP as well as TONEMAP) that needs to be acted upon in a FORM or MENU in an XML Page must be placed above the FORM or MENU. The execution of the tags in an XML Page is sequential in nature. One should not expect a map defined at the end of the Page to have an effect on the tags executed in that Page. If no tone is entered by the caller within the timeout period specified in the PAUSE sub-element of PLAY in the above MENU, a TIMEOUT exception is raised and control goes to the corresponding HREF (in this example to the A label with NAME="label1"). If an invalid key is entered, TONEID=OTHER applies and control again goes to label1. REPEAT="MENU" could have been used instead of HREF=#label1 to achieve the same effect in the TIMEOUT EXCEPTION.

In the following Example 3, an XML Page for accepting a set of input fields (a Social Security Number and Password) is presented. A FORM element allows a caller to enter multiple FIELDs before the data is sent over to the call center. For example in the following FORM data for both the ssnum and passw FIELDs is collected before it is sent (when the end tag </FORM> is encountered).

```
<-- Example # 3-->
<?xml          version="1.0" ?>
<XMLPage       TYPE= "IVR" CUSTID="customer"  PAGEID="v102"
VERSION="1.7"
               SESSIONID=" 3"
        HREF="www.customer.com/tele/operator.asp?SESSIONID=3" >
<FORM                  METHOD="GET"  HREF="www.customer.tele/cgi-
bin/validate.asp?SESSIONID=3" >
<FIELD                 HREF="www.customer.com/tele/ss.asp?SESSIONID=3"
MAXREPEATS="3">
<PLAY>
<VOICE         SRC="www.customer.com/tele/ss.vox" TEXT="Please enter your 9 digit
               social security number"/>
<PAUSE         TIME="10" />
</PLAY>
<INPUT         NAME="ssnum"  MINDIGITS="9" MAXDIGITS="9"
MAX_IDD="2/>
<EXCEPTIONMAP >
<EXCEPTION EVENT="TIMEOUT" REPEAT="FIELD" />
```

```
<EXCEPTION EVENT="TOOFEWDIGITS"
HREF="www.customer.com/tele/invalid.asp?SESSIONID=3" />
</EXCEPTIONMAP>
</FIELD>
<FIELD                HREF="www.customer.com/tele/passw.asp?SESSIONID=3"
MAXREPEATS="3">
<PLAY>
<VOICE      SRC="www.customer.com/tele/passw.vox" TEXT="Please enter your 3-
            7 digit password followed by the pound sign"/>
<PAUSE      TIME="5" />
</PLAY>
<INPUT      NAME="passw"  MINDIGITS="3" MAXDIGITS="8"  MAX_IDD="2"
ACCEPT="#"  />
<EXCEPTIONMAP >
<EXCEPTION EVENT="TIMEOUT" REPEAT="FIELD" />
<EXCEPTION EVENT="TOOFEWDIGITS" REPEAT="FIELD" />
</EXCEPTIONMAP>
</FIELD>
</FORM>
</XMLPage>
```

Any values input by the caller for the various fields in the FORM are sent as a query-string (URL followed by ? and NAME, VALUE pairs) to the URL specified by the HREF attribute in the FORM. This query-string is sent when the end tag </FORM> is encountered. The MAX_IDD="2" attribute in INPUT specifies that if no key is entered for 2 seconds since the last key was entered, it is considered that the caller has finished entering the input value as if an ACCEPT key was entered. (IDD is short for Inter-digit-delay). The distinction between a TIMEOUT event and MAX_IDD is rather subtle. The EXCEPTION with EVENT= TIMEOUT is encountered if not even the first digit is entered within the period specified by PAUSE's TIME attribute, after the voice file has stopped playing. However, if one or more digits are entered within TIME seconds and then no digit is entered for MAX_IDD seconds, the field input is considered to be complete. MAXDIGITS for the password field is specified as 8 even though the max password length is 7. This is to accommodate the # key which is the ACCEPT character.

In the following Example 4, an XML Page for playing back a caller's account record (e.g., an account balance) is provided.

```
<!--Example # 4-->
<?xml       version="1.0" ?>
<XMLPage    TYPE= "IVR" CUSTID="customer"  PAGEID="0004"
VERSION="1.7"
```

```
                SESSIONID=" 3"
        HREF="www.customer.com/tele/next.asp?SESSIONID=3" >
<PLAY           INTERRUPTIBLE="NO" >
<VOICE          SRC="www.customer.com/tele/balance.vox"  TEXT="Your account
                balance is" />
<PAUSE          TIME="1" />  <!-- default value of UNIT is "SEC".  You need not
                include it if time is in seconds >
<CURRENCY  SRC=" www.customer.com/tele/currency.indexfile" VALUE="7568.29"
                COUNTRY="USD" />
<PAUSE          TIME="1" />
</PLAY>

<!-- Now ask the caller to make a choice from a MENU for going back to the main menu
or ending this call>

<MENU>
<PLAY>
<VOICE          SRC="www.customer.com/tele/end.vox" TEXT="Press 1 to go back to
                the main menu, 2 to end this call"  />
<PAUSE          TIME="5" />
</PLAY>
<TONEMAP >
<TONE           TONEID="1"
HREF="www.customer.com/tele/main.asp?SESSIONID=3" />
<TONE           TONEID="2"
HREF="www.customer.com/tele/end.asp?SESSIONID=3"  />
<TONE           TONEID="OTHER"  REPEAT="MENU"  />   <!-- repeats the prompt
                if any other key is entered -->
</TONEMAP>
</MENU>         <!-- no EXCEPTIONMAP specified here because the global
        EXCEPTIONMAP is
                adequate >
</XMLPage>
```

In the following Example 5, an XML Page for playing a goodbye message and transferring control back to the CFM for ending this call is provided.

```
<!-- Example # 5 -->
<?xml           version="1.0" ?>
```

```
<XMLPage      TYPE= "IVR" CUSTID="customer"  PAGEID="0004"
VERSION="1.7"
              SESSIONID=" 3
                    HREF="www.customer.com/tele/CFC.asp?SESSIONID=3&amp
              ; CC.NEXTACTION=END_CALL" >
              <!--above query-string tells the Call Flow Manager to end this call -->

<PLAY         INTERRUPTIBLE="NO" >
<VOICE        SRC="www.customer.com/tele/goodbye.vox"  TEXT="Goodbye and
              have a nice day" />
</PLAY>
</XMLPage>
```

Example 6, below, provides an XML page for directing the CFM to transfer a call to an operator.

```
<!-- Example # 6 -->
<?xml         version="1.0" ?>
<XMLPage      TYPE= "IVR" CUSTID="customer"  PAGEID="0004"
VERSION="1.7"
              SESSIONID=" 3"
        HREF="www.customer.com/tele/CFC.asp?SESSIONID=3$&amp;
              CC.NEXTACTION=MAKE_OUTCALL&amp;TELNUM= (408)730-
        8647" >
              <!--above query-string tells the Call Flow Manager to make an outbound
        call -->
<PLAY         INTERRUPTIBLE="NO" >
<VOICE        SRC="www.customer.com/tele/transferring.vox"  TEXT="You are
              being transferred to an operator" />
</PLAY>
</XMLPage>
```

In addition to interaction between the IVR application and the POP gateway, there is also interaction between the CFM and the IVR application. As mentioned above, the CFM is a software component which executes on a server in the VIN and facilitates communication between the call center IVR application and the POP gateway server. There need not be any direct interaction between the CFM and the IVR application (and whether they reside on the same computer platform or different platforms is not important). Instead, these applications may communicate directly with the POP server. When control needs to be transferred from the CFM to the IVR application, the CFM sends an XML Page to the POP server with the next HREF pointing to the IVR application's URL. Similarly when the IVR application needs to send control

back to the CFM, it directs the POP server to request the next XML Page from a URL associated with the CFM. Any parameters that need to be passed are sent as name/value pairs in query-strings after the URL name in the HTTP request. This is explained in more detail with the help of specific examples below:

Consider a situation where an incoming call has just been received at a POP server. With the help of a central lookup database the POP server learns the customer's (i.e., the call center provider's) CFM URL corresponding to the called number. The POP server now sends a HTTP request to the following URL:

http://<call-mgr-

url>?DID=<did>&ANI=<ani>&CC.NEXTACTION=PROCESS_INCALL

In this example, <call-mgr-url> is the Call Flow Manager URL obtained from the lookup server (and does not include the query string); <did> is the called number for the incoming call; <ani> is the caller's number for the incoming call; and CC.NEXTACTION=PROCESS_INCALL indicates that the next action required of the CFM is to process a new incoming call.

In response to the above, the CFM may return an XML Page to the POP server indicating whether the call should be accepted or rejected (depending upon the availability of certain resources). If the call is to be accepted, the URL of the customer's IVR application is included in the (ACCEPT_CALL) XML Page, together with a SESSION ID to be included in all future communications between the POP server and the customer's IVR application. The POP server's first HTTP request to the customer's IVR application would then have the following form:

http://<ivr-url>?DID=<did>&ANI=<ani>&SESSIONID=<sessionid>

In this example, <ivr-url> is the IVR application's URL sent to the POP server by the CFM in the ACCEPT_CALL page; <sessionid> is the SESSIONID sent to the POP server by the CFM in the ACCEPT_CALL page; <did> is the called number for the incoming call; and <ani> is the caller's number in the incoming call

From this point forward, the IVR application is in control and directs what prompts and choices the POP server must present to the caller with the help of XML Pages (such as those provided in Examples 1 - 6 above) with the appropriate tags. The SESSIONID is included as part of the query-string for each HTTP request from the POP server and as an attribute in the XML Page tag in each XML Page sent by the IVR application. DID and ANI need not be included in the query-string of any HTTP request from the POP server except the first.

When the IVR application is ready to turn control back to the CFM for ending the call or for making an outbound call, the IVR application may send the POP server an HREF with the following value:

http://$call-mgr-url$?SESSIONID=$sessionid$&CC.NEXTACTION=END_CALL

The only NextAction requests that need to be transmitted to a CFM are NEW_SESSION_REQ and END_CALL. NEW_SESSION_REQ is an interrupt is sent by a Web application (e.g., Click-to-talk) to a CFM to prompt the CFM to start a new session. The parameters for this request are:

IVRURL: The URL that should be part of a subsequent CREATE_LEG_REQ interrupt to a POP server. This tells the POP server from where it should fetch its next XML Page after a new conversation has been started.

TELNUM is used to help the CFM identify the POP server (and gateway) on which the new conversation should be started.

NUM_LINES specifies the number of outbound telephone lines that need to be reserved for this session. For click-to-talk applications, this number will always be 2. The second phone number needed in a click-to-talk application should be carried inside the customer application's IVRURL.

APP_NAME identifies the application for which this action is desired. This helps the CFM front-end identify the CFM process to which this request should be presented.

FAILURE_URL specifies the URL that should be used to inform components of the VIN that a failure occurred in executing the NEW_SESSION_REQ.

The response to the NEW_SESSION_REQ interrupt should be an XML Page containing a single RESPONSE tag with a RESULT attribute indicating success or failure. The HREF attribute in the XML Page header should be null ("").

An END_CALL query string may be sent by a POP server to a CFM in response to either a normal hang-up by the IVR application/caller or an error situation that cannot be otherwise handled (e.g., such as when the IVR server goes down) and there is no option except to end the call. If the IVR application wants to end the call, the last XML Page will have an HREF pointing to the CFM and it will have the following query string: CC.NEXTACTION=END_CALL &SESSIONID=$sessionid$ &REASON=NORMAL If the caller hangs up in the middle of the IVR session, the POP server will first send an HTTP request to the CFM with the following query string: CC.NEXTACTION=END_CALL plus other name/value pairs. The POP server will also send an HTTP request to the IVR application at a URL specified as part of CALLERHUP global EXCEPTION. If the IVR server goes down for some reason, the POP server may send the following query string to the CFC, depending upon the value set in the IVR_ERROR exception of the CC_EXCEPTIONMAP (the POP server should already have set the values of $last-error$, $error-url$ and $last-error-string$ by then): CC.NEXTACTION=END_CALL&REASON=IVR_ERROR&SESSIONID=$sessionid $&ERROR=$last-error$& URL=$last-error-url$&DESCRIPTION=$last-error-string$.

As was the case for communications between the POP server and the IVR application, the basic method of communication between the POP server and the CFM is through the use of XML Pages. This communication starts as soon as an incoming call is presented at the POP server, and, through a DID-to-CallMgrURL translation using a central lookup server at a Network Operations Center (NOC) (see Figure 9), the POP server learns the URL of the CFM. The POP server sends information about the request in the form of NAME, VALUE pairs in the query-string of the XML Page request. The CFM returns responses in the form of XML Pages with appropriate elements. The following elements/tags are available to the CFM for such call control responses. In addition, the LEG_WAIT tag introduced above may also be used for communications between the POP server and the CFM. So too are the BRIDGE_CALL and UNBRIDGE_CALL tags used. The only difference for the CFM is that it cannot use a LEG_ID of ALL. It must specify one and only one LEG_ID explicitly to bridge and/or unbridge the call.

An "XMLPage" element is the root element for every XML Page used by the CFM for communication with the POP server. It can have the following attributes:

TYPE	can take on a value of CC or IVR. The CFM uses value CC (call control).
CUSTID	identifies the customer
PAGEID	(optional) identifies the page itself
VERSION	used by the customer for identifying the version number of his application
SESSIONID	an identifier for identifying the particular session with the caller

Each XMLPage represents a response to a call control request by the POP server. The possible children (sub-elements) of XMLPage are: ACCEPT_CALL; LEG_WAIT; RESPONSE; DIAL_CALL; BRIDGE_CALL; UNBRIDGE_CALL; HANGUP_CALL; DESTROY_LEG; REJECT_CALL; END_CALL; MAKE_OUTCALL; SET; CC_EXCEPTIONMAP; EXCEPTION; and IMMEDIATE_TRANSFER.

The ACCEPT_CALL element is used by the CFM to alert the POP server that the resources are available for an incoming telephone call and that the POP server should accept the call. The attributes of this element are:

RINGS	(optional) An attribute to control the number of rings in the ringback before answering the call. As rings as necessary are played until the first XML

Page from the IVR application is received.

UNIT	(optional) value can be SEC or NUMBER to indicate the units of RINGS.
HREF	URL of the IVR application from which the next XML Page should be fetched by the POP server after it has answered the call.

This XML Page tag is sent in response to a request from the POP server when an incoming call comes in. The request to the URL of this page should contain the DID and, if available, the ANI of the incoming call as part of the query-string. The ACCEPT_CALL response page returns a SESSIONID, which is the unique identifier for the session created by the CFM. All future requests from the POP server to the CFM (as well as the IVR application) should carry the SESSIONID instead of the DID and ANI. ACCEPT_CALL does not have any sub-elements as children.

The XML Page containing the ACCEPT element (the very first page from the CFM) should also include a CC_EXCEPTIONMAP containing an IVR_ERROR_EXCEPTION to handle non-recoverable IVR errors. The CC_EXCEPTIONMAP is a global map and should be placed before any ACCEPT tag.

The RESPONSE tag is sent by the CFM or a POP server in response to an interrupt request (see above). It indicates whether the action requested by the interrupt was successful or not, and if a failure the reason for the failure. RESPONSE has two required attributes and no children.

RESULT	SUCCESS or FAILURE.
REASON	(optional) - in case of FAILURE, the possible values are UNKNOWN_REQ, INVALID_PARAMETERS, NO_RESOURCES

An XML Page with a DIAL_CALL tag may be transmitted by the CFM in response to a request by the IVR and other applications to make an outbound call. The attributes of DIAL_CALL are:

TELNUM	telephone number that the POP server needs to call
ANI	(optional) the ANI that telephony manager should send in the outgoing call. This value is

	specified by the application in QUEUE_CALL tag

If the outbound call fails, the POP server immediately informs the CFM by sending a request to the HREF specified in the EVENT=DIAL__ERROR exception. If the outbound call succeeds, the POP server falls through to the next tag (if there is one) or to the HREF in the XML Page tag at the top of the Page (if DIAL_CALL is the last tag on the Page. DIAL_ CALL has one child element – CC_EXCEPTIONMAP. Also a global CC_EXCEPTIONMAP should be specified with CALLERHUP event. $end-call-url$ should also be set for the conversation controller by this time.

A HANGUP_CALL element is used by the CFM to inform the POP server that the call on the leg receiving this tag should be terminated. The attributes of this element are:

REASON	(optional) NORMAL or ERROR or CALLERHUP.

The HANGUP_CALL tag differs from the END_CALL tag in that it does not destroy the conversation controller associated with the dropped call. The HANGUP_CALL tag may be followed by a DESTROY_LEG tag if no more calls are to be made on the associated conversation controller or it may be followed by another DIAL_CALL to make another outbound call. HANGUP_CALL does not have any sub-elements.

The DESTROY_LEG element is used by the CFM to inform the POP server that a conversation controller should be terminated. The attributes of this element are:

REASON	(optional) NORMAL or ERROR or CALLERHUP.

DESTROY_LEG does not have any sub-elements. Even though it is expected that DESTROY_LEG would be preceded by a HANGUP_CALL tag if there is an active call on the leg, in case the conversation controller does receive a DESTROY_LEG without a preceding HANGUP_CALL then it should do the necessary cleanup needed to destroy this leg.

A REJECT_CALL element is used by the CFM to inform the POP server that the resources for the call have not been allocated and that the POP server should reject the incoming call by presenting a busy signal. The attributes of this element are:

REASON	(optional). NOPORTS or OTHER.

REJECT_CALL does not have any sub-elements.

An END_CALL element may be used by the CFM to inform the POP server that the call should be terminated. An XML Page with this tag is sent in response to a request to the CFM

from the POP server to end the call. The POP server uses CC.NEXTACTION=END_CALL in the query string of such a request to so advise the CFM. The attributes of this element are:

REASON	(optional) NORMAL or IVR_ERROR or CALLER_REQ.

This tag is also sent back by the CFM to the POP server when an inbound or outbound call has already terminated (see MAKE_OUTCALL below). The CFM cleans up and decrements its resource usage count after sending this tag to the POP server. END_CALL does not have any sub-elements.

An XML Page with a MAKE_OUTCALL tag is sent by the CFM in response to a request by an IVR application to make an outbound call and bridge the caller's incoming call to this outbound call. This request is made by the POP server on behalf of the IVR application by following the HREF on an XML Page sent by the IVR application. For example, if the IVR application wants to connect the caller to an operator when he/she presses the key "0", the last XML Page sent by the IVR application should have an HREF corresponding to TONEID=0 with the following value:

http://$call-
mgrurl$?SESSIONID=$sessionid$&CC.NEXTACTION=MAKE_OUTCALL&
TELNUM=<tel-num>,

where <telnum> is replaced by the telephone number that the IVR application wants the POP server to call and $sessionid$ is replaced by the session ID that was created when the inbound call first came in.

The attributes of MAKE_OUTCALL are:

TELNUM	telephone number that the POP server needs to call
HREF	call flow conductor URL to notify (when the outbound finally ends) if the call is successful

If the outbound call fails, the POP server immediately so informs the CFM by sending a request to the HREF specified in the EVENT=OUTCALL_ERROR exception (see example 9 below). If the outbound call succeeds, the POP server does not send any notification to the CFM immediately. Rather, it waits for the outbound call to be finished (either by the caller or the called party), before sending a request to the HREF attribute of MAKE_OUTCALL. The CFM sends back an XML Page with an END_CALL tag in response to both the success and failure notifications.

MAKE_OUTCALL has one child element: CC_EXCEPTIONMAP, which specifies a mapping between certain call control exceptions and the URLs of the XML Pages to fetch if such an exception happens. A CC_EXCEPTIONMAP with the IVR_ERROR EXCEPTION should be

included before any ACCEPT tag in the first XML Page from the CFM. A CC_EXCEPTIONMAP with the OUTCALL_ERROR EXCEPTION should be included within the MAKE_OUTCALL and IMMEDIATE_TRANSFER tags.

The EXCEPTIONMAP may use the child element EXCEPTION to specify the mapping between an exception and the URL from which to fetch the next Page, in case such an exception is encountered. EXCEPTION has two required attributes:

EVENT	Possible values are BRIDGE_ERROR, UNBRIDGE_ERROR, IVR_ERROR and OUTCALL_ERROR.
HREF	The URL from which to fetch the next page in case the exception specified by EVENT is encountered.

EXCEPTION has no children.

IVR_ERROR indicates an unrecoverable IVR application error. The first XML Page from the CFM (that includes an ACCEPT tag) should include a CC_EXCEPTIONMAP with the IVR_ERROR event and this should be placed before the ACCEPT tag. OUTCALL_ERROR indicates that the attempt to dial a call failed. An XML Page with the MAKE_OUTCALL or IMMEDIATE_TRANSFER tag should include a CC_EXCEPTIONMAP with the OUTCALL_ERROR event.

A SET tag allows the CFM to set the value of a variable for later substitution by the POP. It is a convenient mechanism to allow the CFM to indicate to the POP server that the POP server should associate a particular value with a particular variable name and that the POP server should substitute the variable name with the value when the CFM uses the variable name later in the same or another XML Page. SET has two required attributes and no children.

VARNAME	The name of the variable.
VALUE	The value to be assigned to the variable.

The IMMEDIATE_TRANSFER element is used by the CFM to redirect an incoming call to an operator by making an outbound call and bridging it to the incoming call. This comes in handy if the IVR application server is down for some reason. The attributes of IMMEDIATE_TRANSFER are:

TELNUM	telephone number that the POP server needs to call
HREF	CFM URL to notify (when the outbound call finally

ends) if the call is successful

If the outbound call fails, the POP server immediately so informs the CFM by sending a request to the HREF specified in the EVENT=OUTCALL_ERROR exception. If the outbound call succeeds, the POP server does not send any notification to the call flow conductor. IMMEDIATE_TRANSFER has one child element – CC_EXCEPTIONMAP.

As before, some examples will help to explain the use of the above tags. In Example 7, an XML Page for directing a POP server to accept an incoming call is presented. This Page is sent in response to a new-call indication from the POP server. This indication is usually the first request sent by the POP server to the CFM for a newly arrived call. After allocating resources for a proxy call to the ACD, the CFM returns this Page to the POP server. Note that the Page contains, in the SESSIONID attribute, the address of the call structure reserved by the CFM. If the CFM finds that it is close to exhausting the available ports on the ACD or IVR, it can increase the number of RINGS that are given in the RINGBACK before the call is answered by the POP server.

```
<--Example 7-->
<?xml          version="1.0" ?>
<XMLPage       TYPE= "CC" CUSTID="customer"  PAGEID="v101" VERSION="1.7"
               SESSIONID=" 0xFF0D2040">
<SET           VARNAME="$sessionid$  VALUE="0xFF0D2040" />
<SET           VARNAME="$ivr-root-dir$" VALUE="http://customer/teleb"
<SET           VARNAME="$ivr-url$"  VALUE=" $ivr-root-dir$/ivr.asp" />
<SET           VARNAME="$voicefile-format$" VALUE="MFF_VOX_M_8MHZ" />
<CC_EXCEPTIONMAP >
<EXCEPTION EVENT="IVR_ERROR"  HREF="$call-mgr-
url$?SESSIONID=$sessionid$&amp;
                    ERROR=$last-error$&amp;URL=$error-
                    url$&amp;DESCRIPTION=$last-error-string$" />
</CC_EXCEPTIONMAP>
<ACCEPT_CALL  RINGS="10" UNIT="SEC"
               HREF=" $ivr-
               url$?SESSIONID=$sessionid$&amp;DID=$did$&amp;ANI=$ani$"/>
</XMLPage>
```

Such a Page may be used as the <ACCEPT_CALL> page returned by the CFM in response to the incoming call alert from the POP gateway as shown in Figure 5.

In the following Example 8, a call termination example is provided. In this case, an IVR application has decided that the caller is done and the call should be terminated. In such a case the last XML Page sent by the IVR application will have an HREF which looks like the following:

$call-mgr-url$?SESSIONID=$sessionid$&CC.NEXTACTION=END_CALL

When the POP server sends the request to the CFM, it will send back the following XMLPage:

```
<--Example 8-->
<?xml          version="1.0" ?>
<XMLPage       TYPE= "CC" CUSTID="customer"  PAGEID="v102" VERSION="1.7"
               SESSIONID=" 0xFF0D2040"  >
<END_CALL  REASON="NORMAL" />
</XMLPage>
```

In the following Example 9, an IVR application has decided that the caller needs to make an outbound call. In such a case the last XML Page sent by the IVR application will have an HREF which looks like the following:

```
$call-mgr-
url$?SESSIONID=$sessionid$&amp;amp;CC.NEXTACTION=MAKE_OUTCALL
&amp;amp;TELNUM= (xxx)xxx-xxxx
```

When the POP server sends the request to the CFM, it will send back the following XMLPage

```
<--Example 9-->
<?xml          version="1.0" ?>
<XMLPage       TYPE= "CC" CUSTID="customer"  PAGEID="v102" VERSION="1.7"
               SESSIONID=" 0xFF0D2040"  >
<MAKE_OUTCALL    TELNUM=" (xxx)xxx-xxxx"
                      HREF= "call-mgr-url$?RESULT=SUCCESS
               &amp;amp;SESSIONID=$sessionid$"  >
<CC_EXCEPTIONMAP >
<EXCEPTION EVENT="OUTCALL_ERROR"
               HREF=" call-mgr-url$?RESULT=FAILURE &amp;amp; SESSIONID=$sessionid$"
       />
</CC_EXCEPTIONMAP>
</MAKE_OUTCALL>
</XMLPage>
```

In the following Example 10, an XML Page for informing a POP server that the call has been placed in the ACD's queue, awaiting an agent with the appropriate skill set to become available, is presented. Such a Page may be sent by the CFM in response to a REDIRECT HTTP request (originally from the IVR server) with a query string containing information about the agent group with the right skill set. Through this XML Page, the CFM instructs the POP server to play a music/infomercial file to the caller while the call is on hold. The CALL_QUEUED element

can optionally contain an indication about how long the queue is and what is likely to be the hold time.

```
<--Example 10-->
<?xml          version="1.0" ?>
<XMLPage       TYPE= "CC" CUSTID="customer"  PAGEID="v101" VERSION="1.7"
               SESSIONID=" 0xFF0D2043" >
<CALL_QUEUED SRC="www.customer.com/teleb/music1.vox"  QLEN="5"  TIME="10"
UNIT="MIN" />
</XMLPage>
```

With the above tags, call flow management operations such as those illustrated in Figures 5 and 7 can be easily implemented. For example, as shown in Figure 5, inbound calls may be intercepted at a local POP gateway close to the point of call origin, so as to reduce long distance charges that might otherwise be incurred. In response, the local POP gateway notifies the CFM of the call , seeking instructions for managing the call. This portion of the call is referred to in the diagram as LEG_ID = 1.

In response to the notification from the POP gateway, the CFM returns an <ACCEPT_CALL> page, directing the POP gateway to the appropriate URL to fetch instructions form the IVR. Following this direction, the POP gateway begins an exchange with the IVR application, which provides instructions for handling the inbound call. For example, the IVR may provide instructions regarding data collection from the user that can be prompted using specified menus, as discussed above.

As the dialog between the IVR and the local POP gateway is in progress, the IVR may instruct the POP gateway to request that a proxy call be placed in the call center's ACD, with the point being to ultimately connect the caller to an operator at the call center. As shown in the diagram this may be done using the CREATE_LEG_AND_DIAL command described above. The specified TELNUM is the number to be dialed to reach the call center. In other cases, this instruction may go to a premises gateway server at the call center.

In either case, the gateway server passes a request to the CFM to initiate the proxy call and, in response, the CFM opens a dialog with the requesting gateway to do so. Once the proxy call has been set up (identified as LEG_ID = 2 in the diagram), and the proxy call is about to be answered the two calls (LEG_ID = 1 and LEG_ID =2) are bridged (in response to a BRIDGE_CALL command from the CFM) as shown.

In addition to allowing for the management of inbound calls, the XML command may also be used to manage outbound calls from a call center, as illustrated in Figure 7. This time, the IVR initiates the process by asking the CFM to establish a new call session for a call to an identified telephone number (npa) nxx-xxxx. In response, the CFM determines an appropriate local POP gateway and begins a call set-up dialog, the POP gateway is instructed too place an outbound call to the specified telephone number, using the DIAL_CALL command. Once this

call is established, the CFM hands over control to the IVR (by passing the POP gateway an appropriate URL from which to fetch a Page) and the IVR can then provide the POP gateway with appropriate called-party interaction instructions.

SYSTEM DESCRIPTION

Having thus reviewed the process flow aspects of the present invention, a system overview may be useful. **Figure 8** is a functional diagram of a virtual intelligent network system in accordance with at least one embodiment of the present invention wherein the end user 116 is connected to the business call center 150 via an originating Local PSTN 106, a Long Distance Network 114 and a terminating Local PSTN 106.

The VIN includes of one or more POP call center gateway servers 146 distributed at one or more points of presence 152 close to the points of call origination. These POP call center gateway servers 146 are connected by one or more call center networks 148 to the Premises call center gateway servers 142 at one or more business call centers 150. The call center networks 148 are connected to one or more applications web application servers 154 which host user specified applications, such as the CFM and IVR applications, on behalf of the multiple business call centers 150. The POP call center gateway server 146 is further connected to a switched or dedicated access public telecommunications network 114 enabling long distance voice communications with connected premises call center gateway servers 142.

A business call center 150 may belong to a plurality of business call centers, distributed over a wide area, and may include of one or more premises call center gateway servers 142, one of which would be dynamically selected by the CFM application at the time of handling of an incoming call at a POP call center gateway.

Referring now to **Figure 9**, POP gateway 166, under the direction of the CFM, intercepts and answers inbound calls at or near the point of call origination. In addition, POP gateway 166 provides automated services with the CFM-specified interactive voice response applications, holds and queues a call until instructed to transfer the call to the premises call center gateway 164 at the call center 150 when appropriate operators are available to service the call, and/or plays music or customized announcements to the caller, while the call is on hold. If the call is queued, the CFM application may issue a command to a premises call center gateway 164 to originate a proxy call at the call center 150 and to monitor the progress of the queued call. The premises call center gateway 164 is selected by the CFM using call specific information such as ANI and DNIS or caller account information that is provided by the IVR application deployed on Web server 154, to intelligently choose the best matching answering resource. Then, when the premises call center gateway 164 so alerts the CFM, the CFM directs the POP call center gateway 166 to route the locally queued call to the premises call center gateway 166 just-in-time before an operator answers the call.

The premises call center gateway 164 responds to the CFM command and generates proxy calls to the premises call center ACD. The premises call center gateway 164 then monitors

the progress of proxy calls within the ACD for operator availability, communicates with the CFM for just-in-time call delivery to the selected operator and bridges the call between the POP call center gateway 166 and the premises ACD.

The network operations center (NOC) 156 may host a database that can be accessed by the CFM and/or other resources of the VIN to lookup called numbers. In this way, intelligent called routing may be provided.

Referring to **Figure 10**, a virtual intelligent network, according to one embodiment, is a virtual private network connecting multiple POP call center gateways 166, one or more premises call center gateways 164, all of which may belong to a single business call center 150, and one or more web application servers 154 that host user-specified applications, such as CFM and IVR application for that business call center, through a call center network 148. A virtual private network offers connection and transport protocols such as Asynchronous Transfer Mode (ATM), Frame Relay or Internet Protocol (IP) for secure and private data communications between connected entities with optional quality of service guarantees.

Referring to **Figure 11**, each POP call center gateway 166 can be part of multiple such call center networks 148, one for each business call center 150 served, all connected to one or more web application servers 154 for the business call centers 150. POP call center gateways 166 use a call center network 148 to connect to corresponding premises call center gateways 164 and to access appropriate interactive voice response applications under the direction of the CFM for a respective business call center.

In the foregoing specification the present invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made to the specific exemplary embodiments without departing from the broader spirit and scope of the invention as set forth in the appended claims. For example, the XML commands described above may be stored on any computer-readable media (e.g., CD-ROM, floppy disk, etc.) and/or may be transported as electrical or other (e.g., light, microwave, etc.) signals on a media interconnecting elements of the VIN.

In addition, one should not lose sight of the overall goals of the present VIN. In essence, the VIN extends an enterprise network (e.g., the business call center discussed above) to the edge of an access network in a manner that is transparent to the enterprise network and independent of the underlying media transport mechanisms. Consider, for example, the network topology depicted in Figure 12. Network 200 includes a number (N) of access networks 202, at the edges of which are provided gateways 204. The gateways 204 may be similar to the POP call center gateways discussed above.

Each access network 202 may serve as a local access network to a number of users 206. For example one or more of the access networks 202 may be local loop network (wireless or wireline), cellular telephone networks (analog or digital), digital networks (e.g., IP, ATM, frame relay, etc.), or other access networks. Through the gateways 204, the access networks 202, may be communicatively coupled to one or more enterprise networks (e.g., business call centers, office

computer networks, intra-nets, etc.) 208, via an interexchange network (INX) 210 and/or an IP network 212. IXN 210 may be a long distance or other telecommunications network operated by an interexchange carrier and may use conventional time division multiplied (TDM) transport mechanisms, analog or digital switched communication transport mechanisms, voice-over-IP, voice-over-ATM, voice-over-frame relay, or any other circuit switched or other underlying transport and/or communication mechanisms to carry voice calls and/or data communications. The IP network 212 may include the CFM server described above.

Through the control of local applications executed at the gateways 204, for example under the control of IVR and/or CFM servers as discussed above, the functionality of the gateways 208 is delivered to the edges of the access networks 202. Thus, users 206 may be provided with services without having to incur expenses associated with transport and/or connection access IXN 210. In addition, multiple gateways 208 (which need not be associated with the same business entity) can share the gateways 204, with each having control over applications to be executed at the gateways 204 through the XML control processes discussed above, because such control processes make use of the IP network 212, these control processes also avoid fees associated with the use of IXN 210. Only when a true connection to the enterprise network is needed (e.g., when an operator becomes available) does a call need to be bridged across the IXN 210 (in any manner, e.g., circuit switched PSTN, voice-over-IP, voice-over-ATM, voice-over-frame relay, etc.).

The present VIN also allows for easy end-to-end call routing. For example, using the XML control processes described above, an enterprise network may provide destination address information to a gateway 204 to allow for transport of an incoming call over an IP or other digital network. Consider that an incoming call will be associated with an POTS dialed number. Although a conventional circuit switched PSTN may make use of this dialed number to route the call, a network based on IP, ATM, frame relay, etc., must have an associated address (e.g., an IP address, VP/VCI, DLCI, etc., as the case may be) in order to route the call. Using the XML control procedures described above, the gateway 204 can be provided with such an address to provide call routing over such a network, perhaps even directly to an IP end-point (e.g., an IP phone). Thus, the present VIN separates call management from media management to efficiently use VOIP (VOATM, VOFR, etc.) as a voice (or other data) transport mechanisms. Accordingly, because of the general nature of present VIN and the multiple applications afforded thereby, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method, comprising managing call flows within a tele/data-comminucations network through the use of commands passed between nodes of the network as extensible markup language (XML) tags.

2. The method of claim 1 wherein managing comprises answering terminating, accepting, routing, initiating, bridging unbridging, conferencing and/or disconnecting.

3. The method of claim 2 wherein at least one of the nodes of the network comprises a Web server hosting a call flow manager application configured to send and receive the XML tags to and from other nodes of the network.

4. The method of claim 1 wherein managing comprises receiving notification of a call at a call flow manager within the network and directing whether the cal should be accepted or not through the use of one or more of the XML tags.

5. The method of claim 4 wherein managing further comprises bridging the call in response to one or more commands issued using some of the XML tags with a second call.

6. The method of claim 5 wherein the second call is a proxy call initiated at a call center in response to one or more commands issued using at least one of the XML tags.

7. The method of claim 5 wherein the second call is initiated at a point of presence close to the second call's termination point.

8. The method of claim 7 wherein the second call is initiated in response to a command issued using at least one of the XML tags.

9. The method of claim 4 wherein if the call is accepted, the call is further managed under the control of an interactive voice response (IVR) application hosted on a computer resource within the network.

10. The method of claim 9 wherein the IVR application manages the call by issuing commands relating to caller interaction operations to be performed at the node of the network at which the call was received using various ones of the XML tags.

11. The method of claim 10 wherein at least one of the commands issued by the IVR application includes information regarding an address at which the node of the network at which the call was received may locate instructions for ending the call.

12. The method of claim 10 wherein at least one of the commands issued by the IVR application includes information regarding an address at which the node of the network at which the call was received may locate instructions for playing messages.

13. A set of computer-readable instructions comprising call flow management commands for use by nodes of a tele-/data-communication network and written in an extensible markup language (XML).

14. The computer-readable instructions of claim 13 as embodied as electrical or other signals transported on a media interconnecting at least two nodes of the network.

15. The computer-readable instructions of claim 13 as embodied on a computer-readable medium.

16. A network comprising two or more nodes configured to provide call management operations by communicating with one another through the use of a set of computer-readable instructions expressed in an extensible markup language (XML).

17. The network of claim 16 wherein the instructions are expressed as a sequence of operations to be performed in furthermore of the call management on individual XML Pages.

18. A method comprising:

directing through one or more user-specified web applications a second call center to service an inbound call;

directing through said applications a first call center to establish a connection therein; and

bridging under the control of said applications the inbound call from the second call center when the connection is established in the first call center.

19. The method of claim 18 wherein the web application server, the local call center and the remote call center are communicating through a data network.

20. The method of claim 18 wherein said remote call center is selected from a plurality of call centers distributed over a wide area.

21. The method of claim 18 wherein said user specified applications are running on a web application server.

22. The method of claim 18 wherein said user specified applications control call flow and voice response behavior of said local call center and said remote call center.

23. The method of claim 22 wherein said user specified applications control call flow behavior of the local call center and the remote call center by selecting the remote call

center to achieve an even distribution of call flow among the plurality of the remote call centers.

24. The method of claim 22 wherein said user specified applications control voice response behavior by selecting interactive voice response based automated service and requesting the local center to execute it.

25. The method of claim 22 wherein said user specified applications control said call flow and voice response behavior through eXtensible Markup Language (XML) messages.

26. The method of claim 22 wherein said user specified applications are call flow management (CFM) and interactive voice response (IVR) web applications.

27. The method of claim 23 wherein said user specified applications are deployed on web application servers hosted at external Internet data centers.

28. The method of claim 18 further comprising the user specified application gathering caller's account information, selecting the remote call center and routing the inbound call to the selected remote call center.

29. The method of claim 18 wherein the local call center is servicing the inbound call by intercepting and answering said call near the point of call origination, providing interactive voice response based automated service, holding and queuing the call until the remote call center is available to service the call.

30. The method of claim 29 further comprising the steps of:

requesting through one or more user specified applications queuing of the call by the local call center;

requesting through one or more user specified applications running of the voice response application during a call hold time; and

interrupting through one or more user specified applications said queued call and instructing local call center to transfer the call to a selected answering resource.

31. The method of claim 30 wherein said user specified applications request local call center to release the connection to the answering resource at one location and connect it at the other location if the call needs to be transferred from the former to the later location.

32. The method of claim 18 further comprising the user specified applications requesting the remote call center to originate a proxy call in the remote call center's queue.

33. The method of claim 32 further comprising the remote call center responding to the user specified applications request to originate the proxy call, monitoring the progress of the proxy call for an operator availability and communicating the same to the user specified applications.

34. The method of claim 33 further comprising user specified applications directing the local call center to transfer the held call to the remote call center upon the remote call center communication of the operator availability.

35. A method comprising:

directing through one or more user specified applications a local call center to originate outbound calls;

directing through said applications a remote call center to establish a connection therein; and

bridging under the control of said applications the outbound call with existing calls.

36. The method of claim 35 further comprising:

specifying through one or more user specified applications a set of alternative long distance numbers;

selecting a first long distance number from the set of alternative long distance numbers;

directing through one or more user specified applications a call center which is local to the first selected number to place a first call; and

selecting other long distance numbers from the set of alternative long distance numbers if the placement of the first call to the first selected number was unsuccessful until a call to one of other selected numbers can be completed.

37. The method of claim 35 further comprising selecting a voice message and directing the call center to deliver said voice message to an end user.

38. The method of claim 37 wherein said voice message provides the end user with an option to be connected to a servicing agent.

39. The method of claim 38 further comprising the end user selecting said option and the user specified applications directing the remote call center to establish connection with the servicing agent and bridging the call to the end user when the connection is established.

40. A virtual intelligent network system comprising:

a plurality of remote call centers that are distributed over a wide area for servicing calls;

a local call center to answer calls; and

a web based application server configured to host user specified applications which direct the local call center to service calls until connection in the remote call center is established.

41. The system of claim 40 further comprising a data network connecting the local call center, the plurality of remote call centers, and the web application server.

42. The system of claim 41 wherein the data network is a virtual private network that provides connection and transport protocols.

43. The system of claim 42 further comprising a network operations center, connected to said data network, discovering, configuring, monitoring and managing the virtual intelligent network using web network management tools.

44. The system of claim 43 further comprising Internet Protocol (IP) telephony, Media Gateway Control Protocol (MGCP) Gatekeeper protocols and virtual private networks for long distance voice communications.

45. A method comprising extending an enterprise application to an edge of a local access network through the use of call control mechanisms provided by the enterprise network to a gateway physically located at the edge of the local access network.

46. The method of claim 45 wherein the call control mechanisms comprises XML tags associated with call control functions.

47. The method of claim 45 wherein the gateway is shared among multiple enterprise networks.

48. The method of claim 45 wherein calls received through the local access network are terminated at the gateway.

49. The method of claim 48 wherein calls are bridged from the local access network to the enterprise network across an inter-exchange network.

50. The method of claim 49 wherein the inter-exchange network is one of a circuit switched long distance telephone network, a voice-over-IP network, a voice-over-ATM network or a voice-over-frame relay network.

51. The method of claim 48 wherein the local access network comprises one of a cellular telephone network, a wireless network or a wireline network.

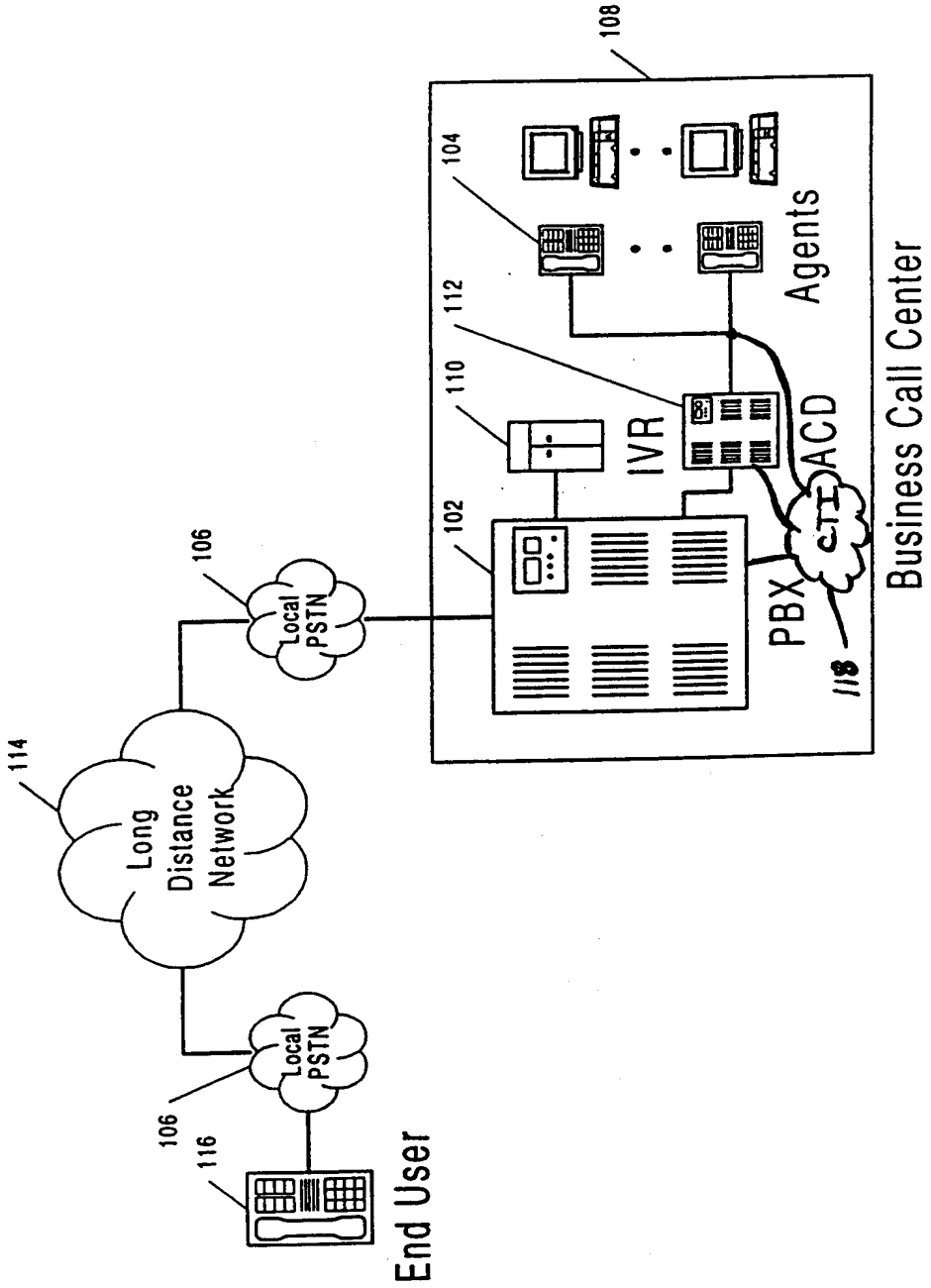


FIG. 1 (PRIOR ART)

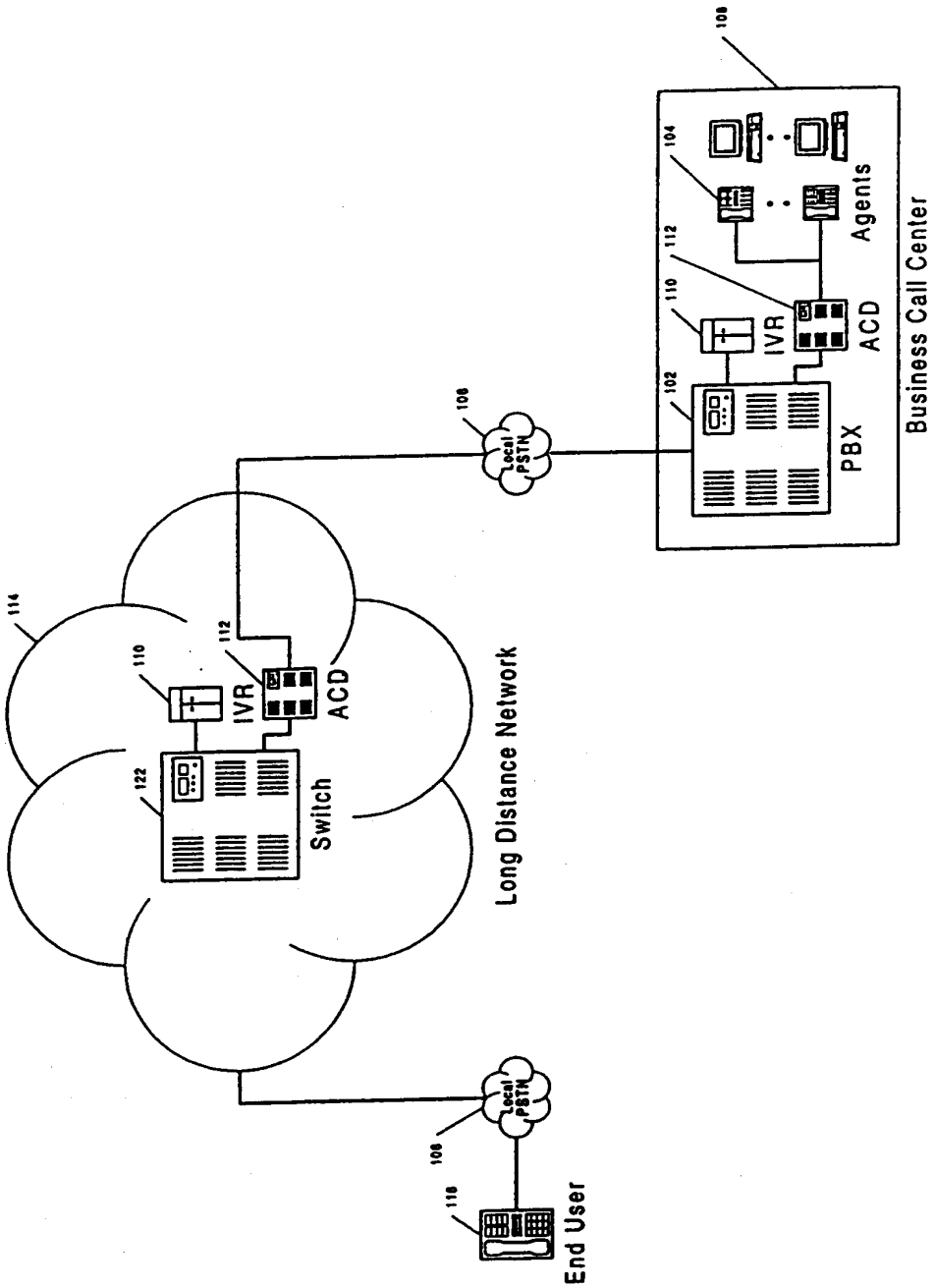


FIG. 2 (PRIOR ART)

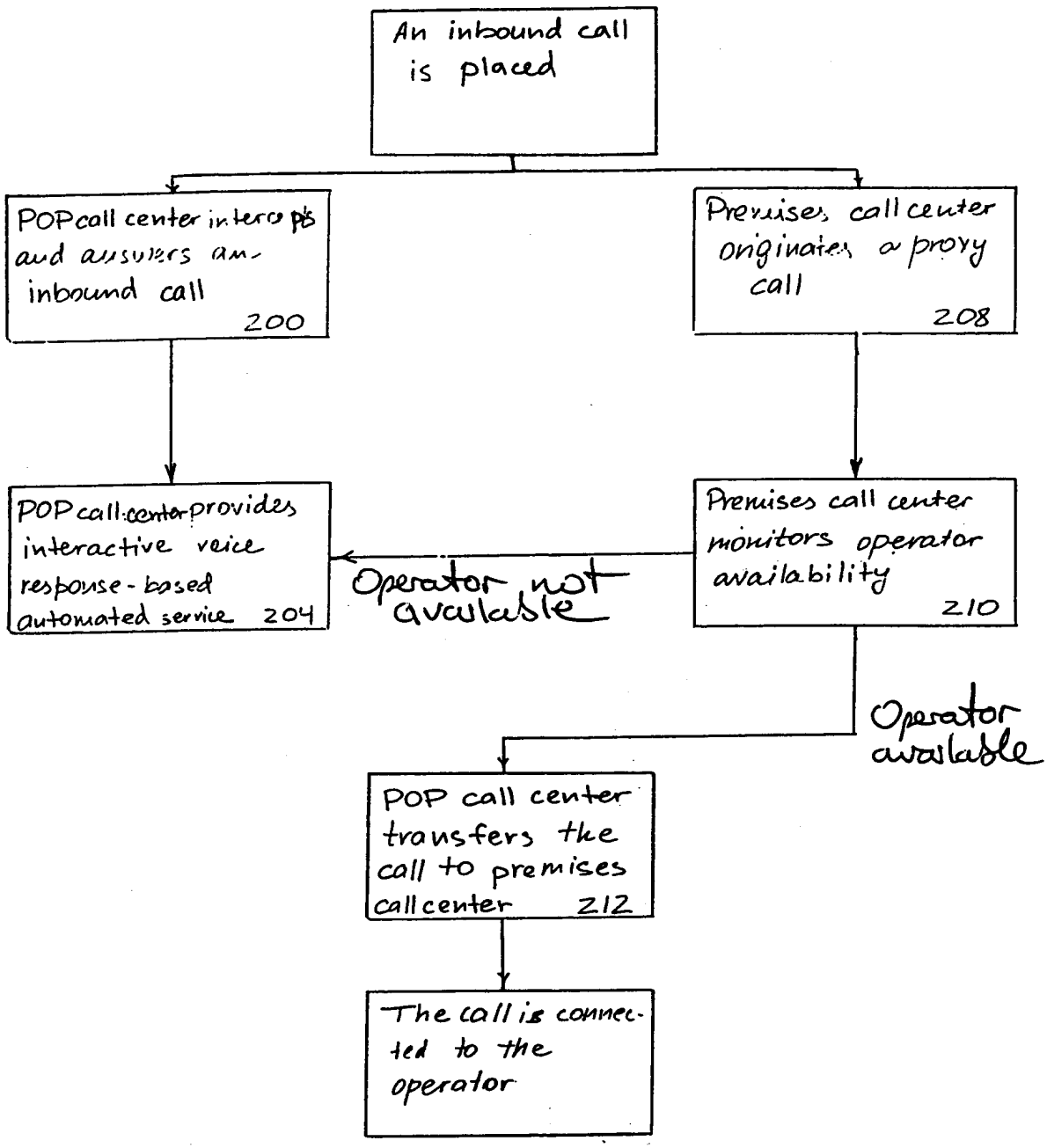

An inbound call is placed
POP call center intercepts and answers an inbound call 200
Premises call center originates a proxy call 208
POP call center provides interactive voice response-based automated service 204
Operator not available
Premises call center monitors operator availability 210
Operator available
POP call center transfers the call to premises call center 212
The call is connected to the operator

FIG. 3

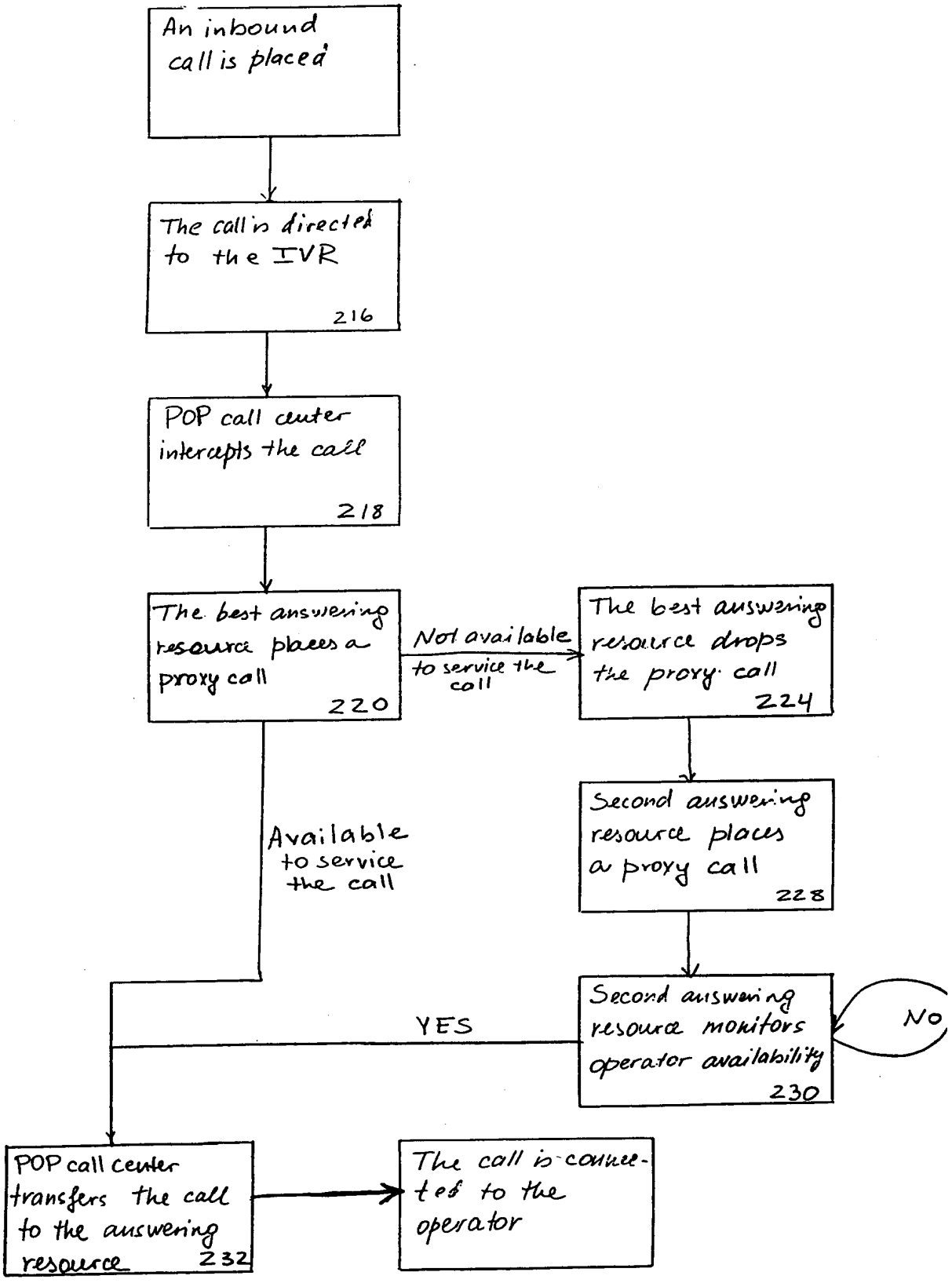

An inbound call is placed
The call is directed to the IVR
216
POP call center intercepts the call
218
The best answering resource places a proxy call
220
Not available to service the call
The best answering resource drops the proxy call
224
Available to service the call
Second answering resource places a proxy call
228
Second answering resource monitors operator availability
230
NO
YES
POP call center transfers the call to the answering resource
232
The call is connected to the operator

FIG. 4

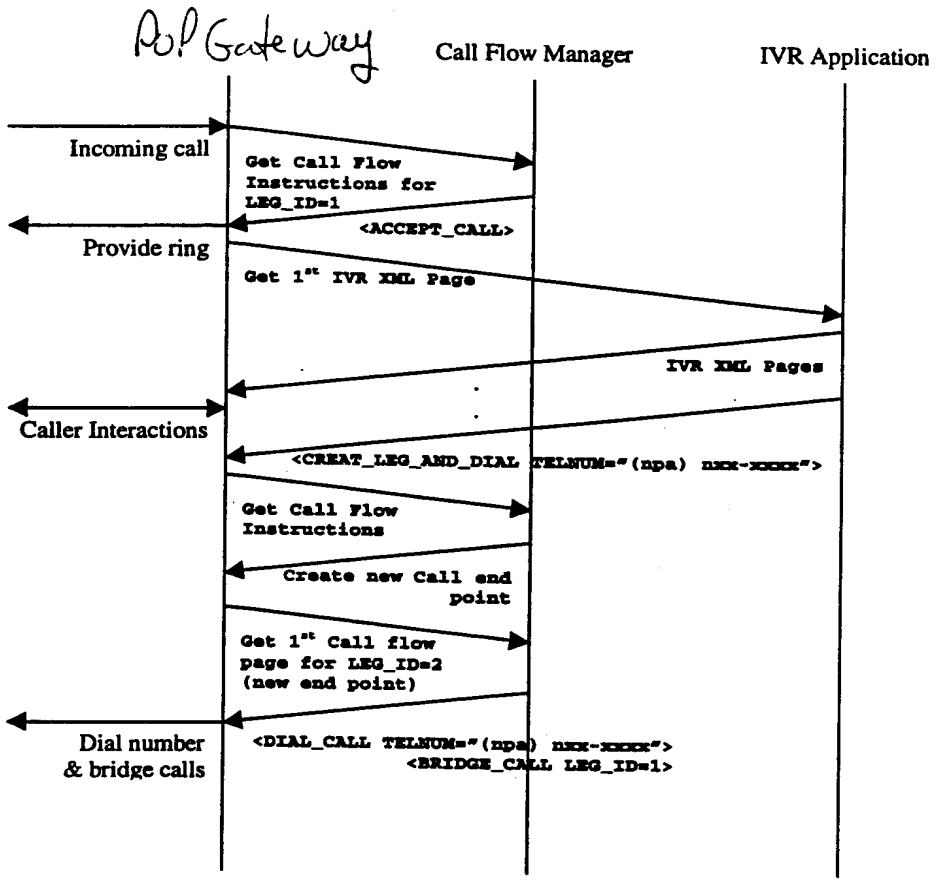
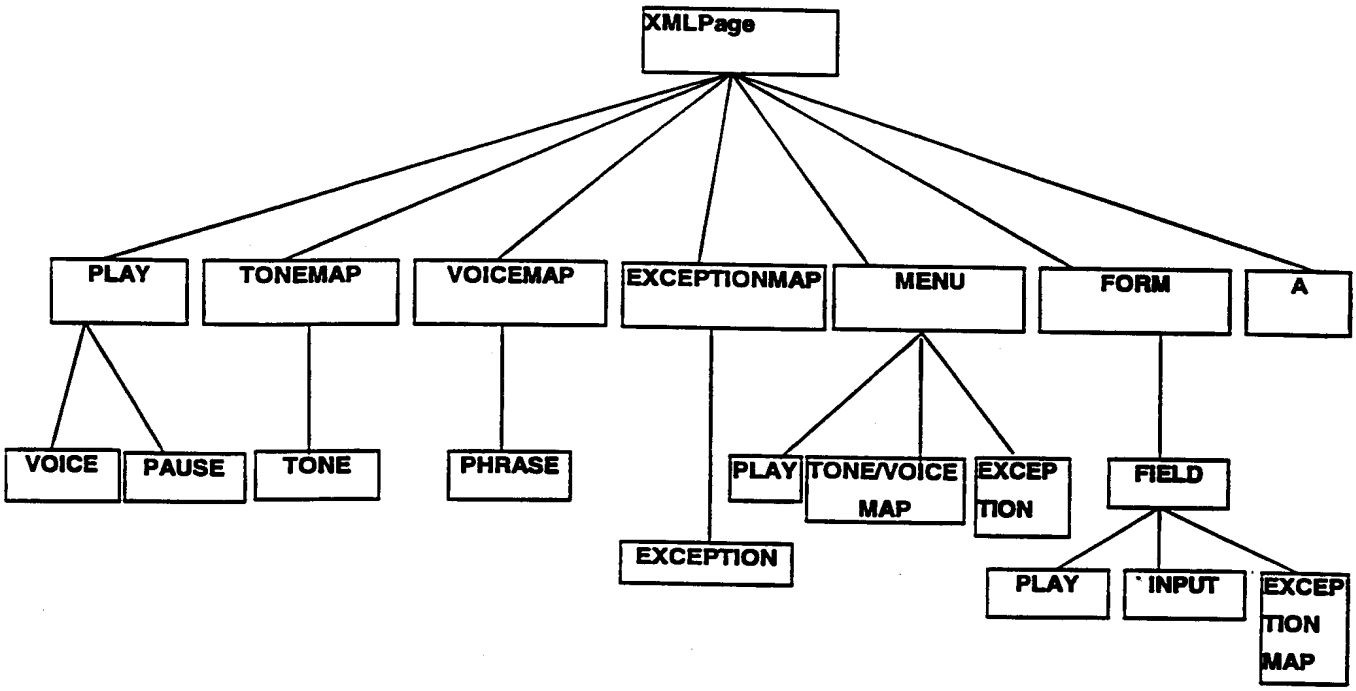

AoL Gateway
Call Flow Manager
IVR Application
Incoming call
Get Call Flow Instructions for LEG_ID=1
<ACCEPT_CALL>
Provide ring
Get 1st IVR XML Page
IVR XML Pages
Caller Interactions
<CREAT_LEG_AND_DIAL TELNUM="(npa) nxx-xxxx">
Get Call Flow Instructions
Create new Call end point
Get 1st Call flow page for LEG_ID=2 (new end point)
Dial number & bridge calls
<DIAL_CALL TELNUM="(npa) nxx-xxxx">
<BRIDGE_CALL LEG_ID=1>

Fig. 5

Fig. 6


XMLPage
PLAY
TONEMAP
VOICEMAP
EXCEPTIONMAP
MENU
FORM
A
VOICE
PAUSE
TONE
PHRASE
EXCEPTION
PLAY
TONE/VOICE MAP
EXCEP TION
FIELD
PLAY
INPUT
EXCEP TION MAP

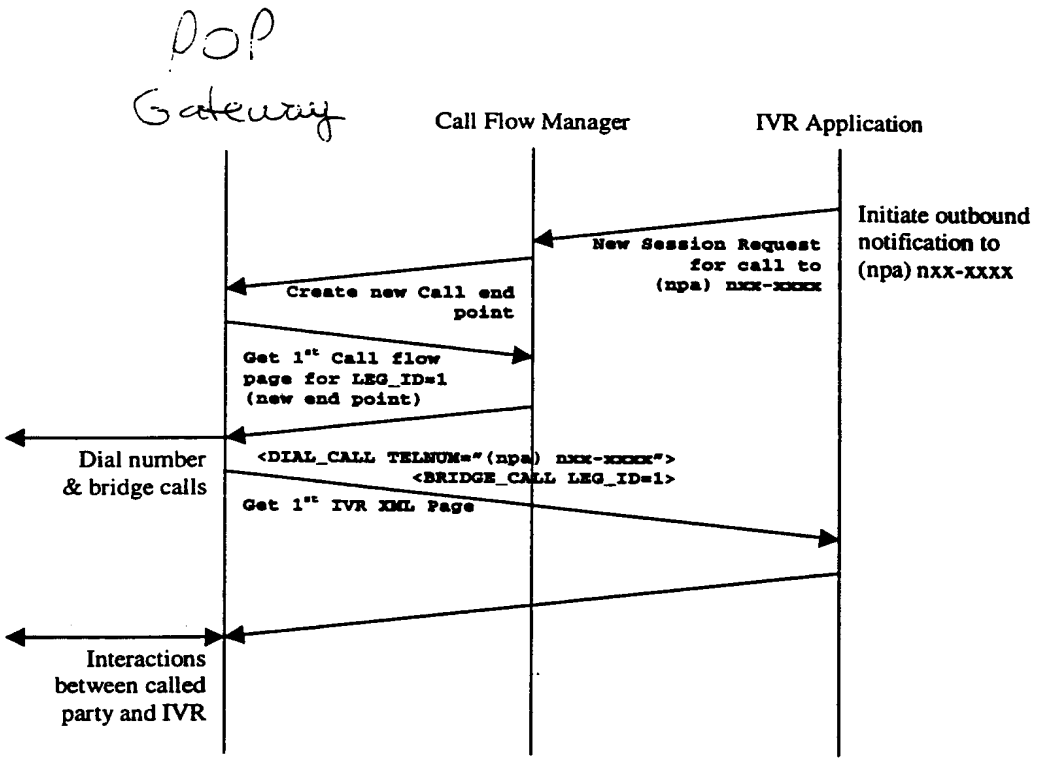

POP
Gateway
Call Flow Manager
IVR Application
Initiate outbound notification to (npa) nxx-xxxx
New Session Request for call to (npa) nxx-xxxx
Create new Call end point
Get 1st Call flow page for LEG_ID=1 (new end point)
Dial number & bridge calls
<DIAL_CALL TELNUM="(npa) nxx-xxxx">
<BRIDGE_CALL LEG_ID=1>
Get 1st IVR XML Page
Interactions between called party and IVR

Fig. 7

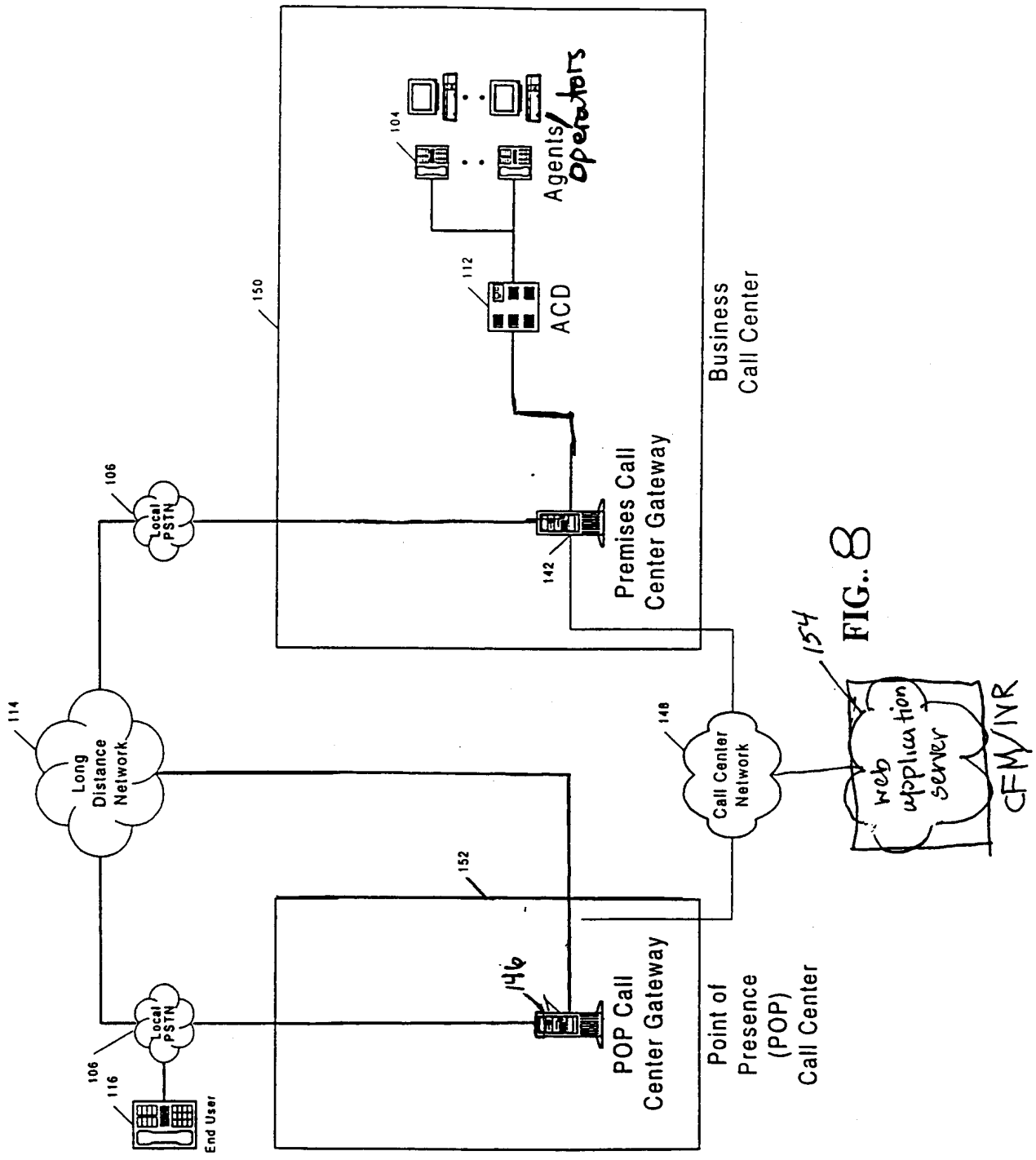

Agents/ Operators
104
112
ACD
150
Business Call Center
106
Local PSTN
142
Premises Call Center Gateway
114
Long Distance Network
148
Call Center Network
154
web application server
CFM/IVR
152
146
POP Call Center Gateway
Point of Presence (POP) Call Center
106
Local PSTN
116
End User

FIG. 8

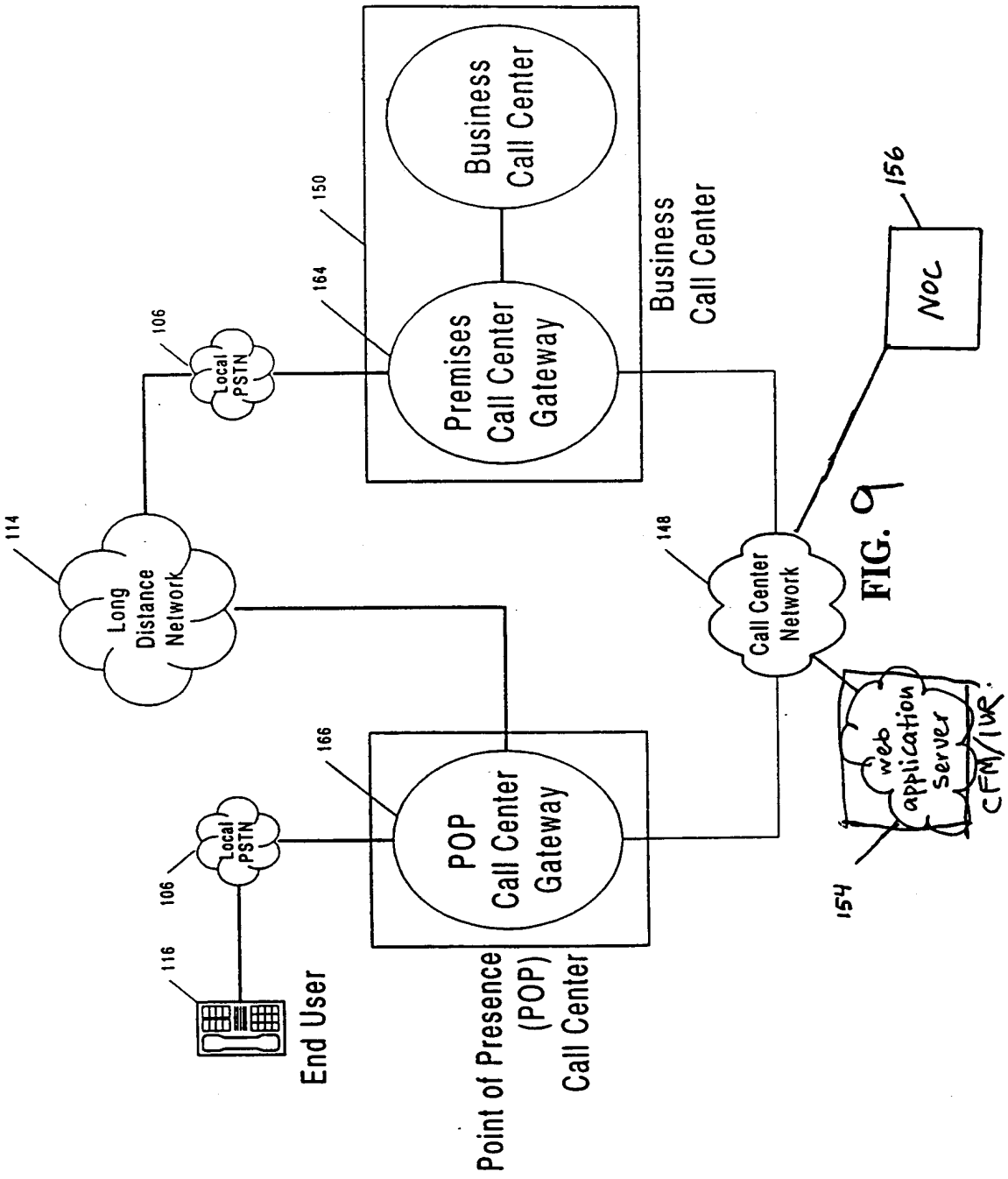

Business Call Center
150
Premises Call Center Gateway
164
Business Call Center
106
Local PSTN
NOC
156
114
Long Distance Network
148
Call Center Network
Web application Server
CFM/IWR
154
166
POP Call Center Gateway
106
Local PSTN
116
End User
Point of Presence (POP) Call Center

FIG. 9

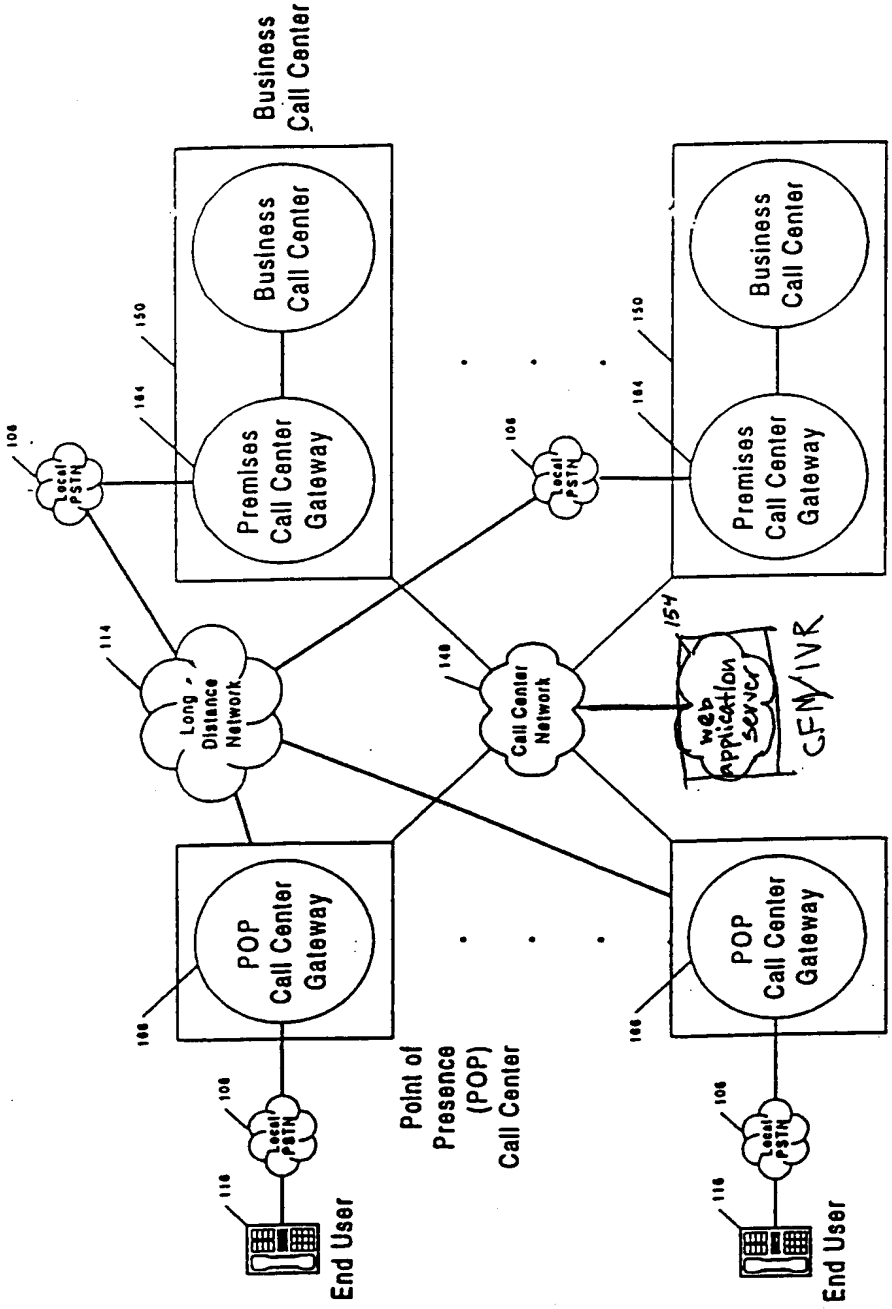

Business Call Center
Business Call Center
Premises Call Center Gateway
Local PSTN
Long - Distance Network
Call Center Network
Web Application Server
CFM/IVR
POP Call Center Gateway
Point of Presence (POP) Call Center
End User

FIG. 10

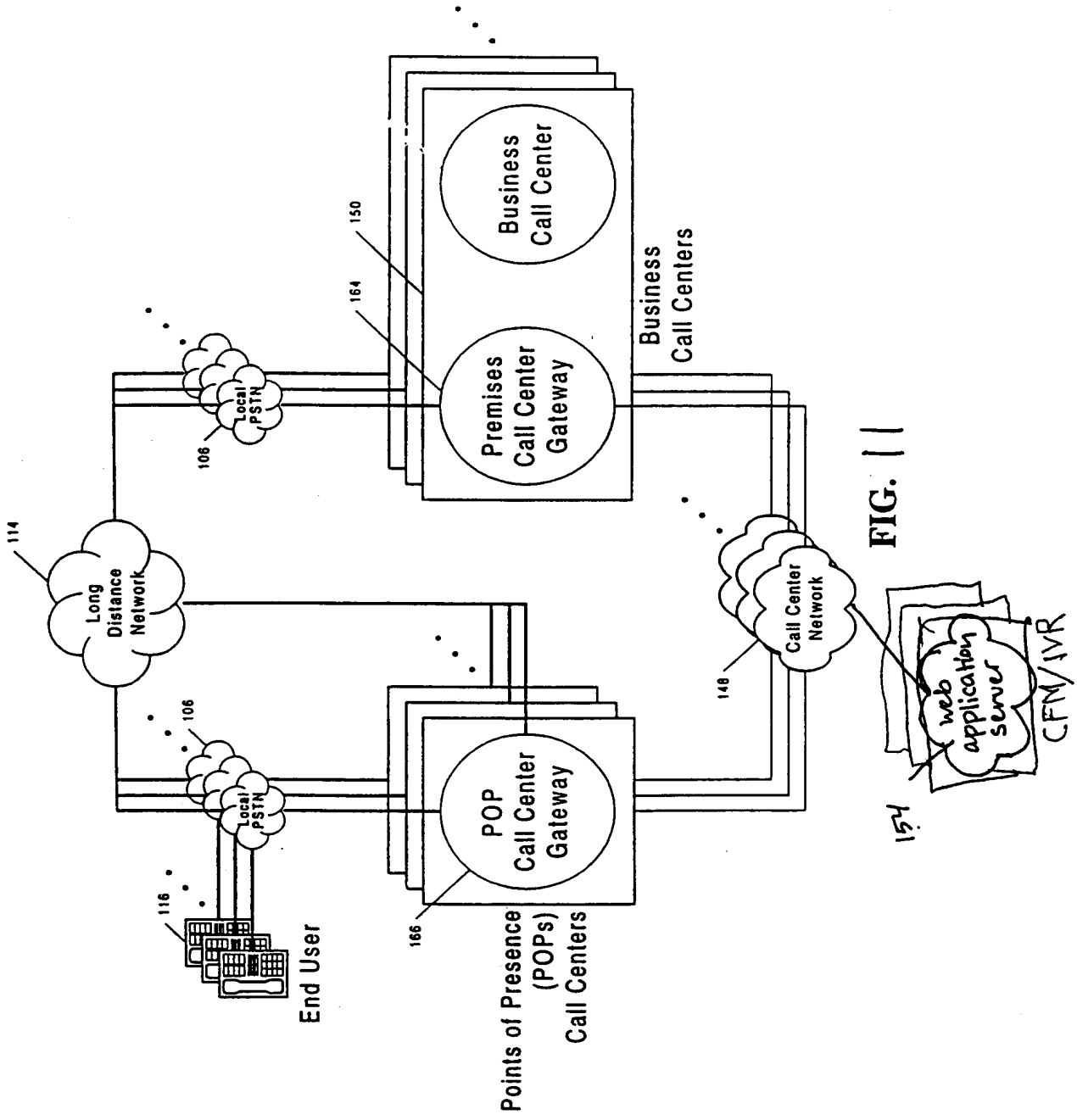

Business Call Center
Premises Call Center Gateway
Business Call Centers
150
164
Local PSTN
106
Long Distance Network
114
Call Center Network
148
web application server
CFM/IVR
154
POP Call Center Gateway
166
Points of Presence (POPs) Call Centers
End User
116

FIG. 11

22-144 200 SHEETS

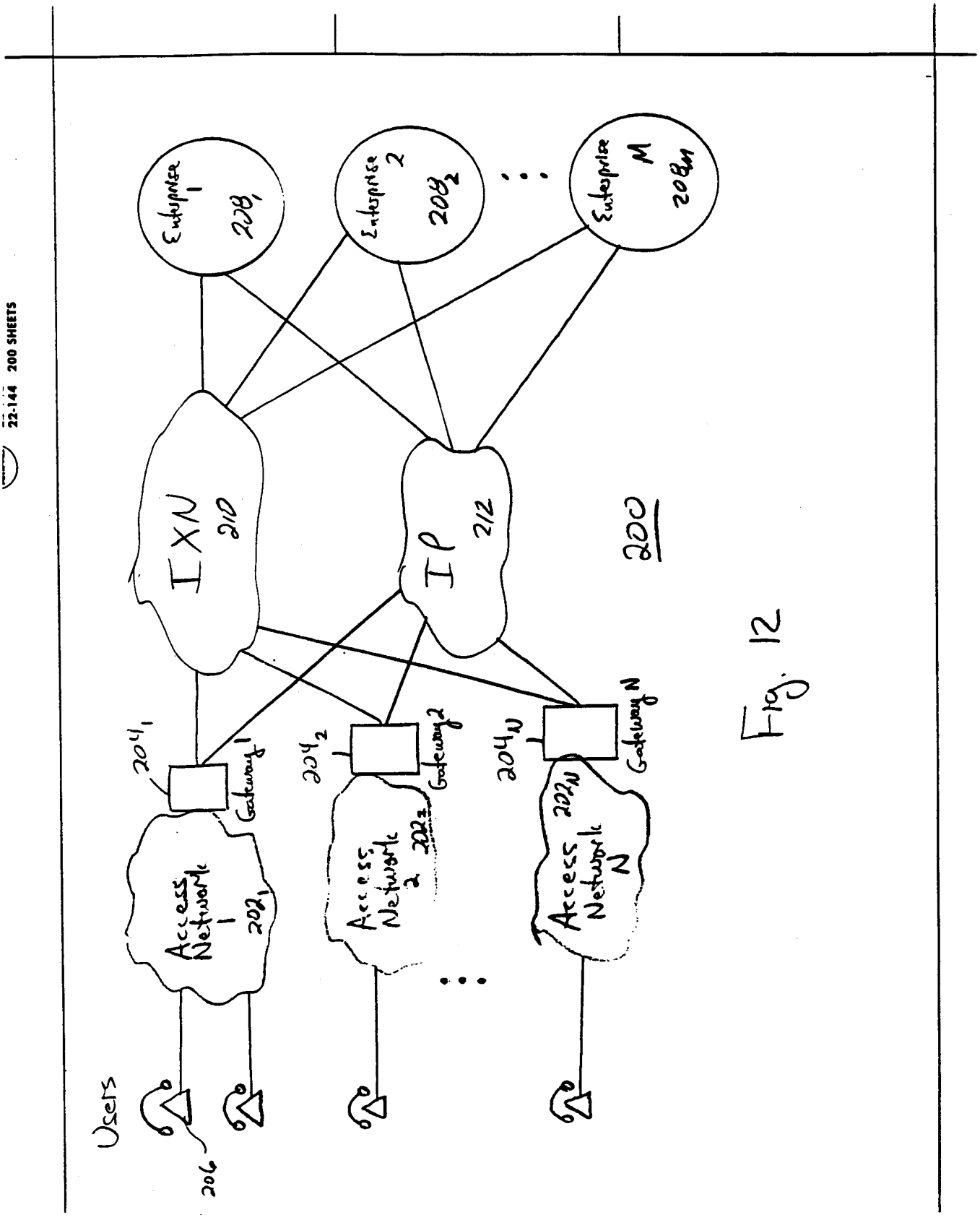


Fig. 12